

Solución de problemas de enrutamiento de vehículos para el transporte de bienes considerando restricciones multidrop e impacto ambiental

Presentado por:

Luis Miguel Escobar Falcón

Presentado como requisito
parcial para optar al título de
Ph.D. en Ingeniería

Director:

Ph.D. Mauricio Granada Echeverri
Universidad Tecnológica de Pereira

Co-director:

Ph.D. John Willmer Escobar Velásquez
Pontificia Universidad Javeriana Cali

Programa de Doctorado en Ingeniería

Línea de Producción

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

Pereira, Julio de 2019



Universidad Tecnológica de Pereira (UTP)

Documento Confidencial

Ni la totalidad ni parte de este documento puede reproducirse, almacenarse o transmitirse por algún procedimiento electrónico o mecánico, incluyendo fotocopias, grabación magnética o electrónica o cualquier medio de almacenamiento de información y sistemas de recuperación, sin permiso escrito de la **UNIVERSIDAD TECNOLÓGICA DE PEREIRA (UTP)**.

Este es un documento interno de la UTP. Al recibirlo no podrá pasarlo a persona alguna excepto las que se le indiquen en la lista de distribución autorizada por la UTP. Cualquier persona externa a la UTP que utilice la información en este documento asume la responsabilidad por su empleo.

Universidad Tecnológica de Pereira (UTP) - 2019

DEDICATORIA

A mi abuelo Jesús y a mi tío José Luis

Luis Miguel Escobar Falcón

Agradecimientos

- A mi hermana Ana María, y a mis padres María del Carmen y Francisco, por el apoyo incondicional y el ejemplo que siempre me han brindado.
- A mi mujer Catalina por su apoyo y comprensión.
- A mi amigo David Álvarez, por compartir sus conocimientos y facilitar el avance por el largo camino recorrido desde la maestría hasta el doctorado.
- A mi profesor y amigo John W. Escobar por su guía y apoyo para culminar este proceso.
- A mi profesor y amigo Rodrigo Linfati por su guía, y a su esposa Carolina por su aliento a culminar este proceso.
- A mi amigo Carlos Contreras y al Grupo de Investigación de Operaciones de la Universidad de Boloña, donde realicé la pasantía, en la cual se generaron importantes resultados para esta tesis.
- A mis amigos César Augusto Marín y Rubén Iván Bolaños por tanto apoyo en la fase final de la investigación, brindándome además todo lo necesario para reingresar al mercado laboral antes incluso de completar el ciclo doctoral.
- A mi orientador Mauricio Granada Echeverri por todo el acompañamiento y confianza depositada al aceptar el reto del proceso doctoral.
- A mi profesor y amigo Anand Subramanian, por su dedicación y consejos para completar el proceso.
- A mi amigo Carlos Adrián Correa, que siempre me apoyó desde la distancia.
- A mis amigos Andrés Domínguez, Andrés Arias, Camilo Acosta, David Escobar y Carlos Saldarriaga, compañeros del programa, por toda la ayuda prestada durante el doctorado.
- Al Profesor Ramón Gallego, por sus consejos, y por haber gestado la maestría y el doctorado que tantas puertas me han abierto.

- A los Profesores Antonio Escobar y Eliana Toro, por todas sus recomendaciones para el desarrollo de la investigación.
- Al Profesor José Germán López Quintero por brindarme su apoyo como Vicerrector de Investigaciones de la Universidad Tecnológica de Pereira, ayudándome en la consecución de la beca doctoral.
- A Marisol Agudelo, amiga y compañera de oficina, quien siempre me ayudó con todas las tareas administrativas del programa.
- A Kenny Cárdenas Parra, amigo, compañero de batallas deportivas, y colega investigador; apoyo vital en la recta final de la tesis y de los retos investigativos en el área de I+D+i de Integra S.A.
- A COLCIENCIAS por la beca proporcionada para realizar el doctorado.
- A Integra S.A. por brindar el espacio ideal para la conclusión de esta investigación y la generación de nuevos proyectos.
- A los profesores miembros del jurado calificador de esta tesis por su atención y disposición a este llamado.

Índice general

Índice general	v
Índice de figuras	x
Índice de tablas	xii
1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Motivación y Justificación del Problema	3
1.3. Objetivos	8
1.3.1. Objetivo General	8
1.3.2. Objetivos Específicos	8
1.4. Estructura del documento	8
2. Marco Referencial	10
2.1. Marco referencial enrutamiento de vehículos y carga de contenedores	10
2.1.1. Enrutamiento de vehículos	10

2.1.2.	Carga de contenedores	13
2.1.3.	Problema integrado de enrutamiento de vehículos y carga de contenedores	14
2.2.	Marco referencial Green Vehicle Routing	19
2.2.1.	Consumo de combustible y modelos de emisión para transporte de bienes por carretera	21
2.3.	Marco teórico	24
2.3.1.	Definición del problema integrado de enruteamiento de vehículos y carga de contenedores	24
2.3.2.	Definición del problema de enrutamiento de vehículos considerando impacto ambiental	28
3.	Algoritmo mateheurístico para el problema de enrutamiento de vehículos con flota homogénea y restricciones de empaquetamiento tridimensionales 3L-CVRP	30
3.1.	Introducción	30
3.2.	Modelamiento del 3L-CVRP	31
3.2.1.	Problema de Enrutamiento de Vehículos Capacitado CVRP	31
3.2.2.	Problema de Carga y Empaquetamiento	34
3.3.	Definición del Problema	35
3.4.	Enfoque Mateheurístico	36
3.4.1.	Método exacto para el Problema de Enrutamiento de Vehículos	38
3.4.2.	GRASP para validación de empaquetamiento	41
3.5.	Resultados Computacionales	42

3.6. Conclusiones y trabajos futuros	48
4. Metaheurística híbrida de Búsqueda Tabú Granular y GRASP para el 3L-CVRP	50
4.1. Introducción	50
4.2. Estructura del Algoritmo Propuesto	50
4.2.1. Solución Inicial	51
4.2.2. Algoritmo GRASP	52
4.2.3. Búsqueda Tabú en Espacio Granular & Algoritmo de Validación 3L tipo GRASP	55
4.3. Experimentos Computacionales	63
4.3.1. Parametrización	63
4.3.2. Análisis Comparativo	65
4.4. Conclusiones e investigaciones futuras	67
5. Algoritmos Mateheurísticos para el Problema del Agente Viajero Considerando Polución	70
5.1. Introducción	70
5.2. Descripción y Formulación del Problema	72
5.3. Algoritmo de Búsqueda Local Iterativa	75
5.4. Algoritmo Genético Multi-operador y Algoritmo Híbrido	79
5.4.1. Representación y función <i>fitness</i>	81
5.4.2. Población inicial	81

5.4.3. Operadores de cruzamiento	83
5.4.4. Operadores de mutación	84
5.4.5. Parámetros del algoritmo genético	84
5.4.6. Algoritmo Híbrido	85
5.5. Experimentos Computacionales	85
5.6. Conclusiones	91
6. Algoritmo genético especializado para el problema de enrutamiento de vehículos con flota heterogénea y restricciones de empaquetamiento bidimensional minimizando el consumo de combustible 2L-FHFVRP	92
6.1. Introducción	92
6.2. Marco Referencial	94
6.3. Descripción del Problema	97
6.4. Estado del Arte	99
6.5. Metodología de Solución	101
6.5.1. Algoritmo GRASP de empaquetamiento bidimensional	104
6.5.2. Algoritmo GA-GRASP	109
6.6. Experimentos computacionales y análisis de resultados	112
6.7. Conclusiones y trabajos futuros	117
7. Conclusiones, trabajos futuros y producción	119
7.1. Conclusiones	119
7.2. Trabajos Futuros	121

7.3. Producción	122
Bibliografía	124

Índice de figuras

3.1. Solución obtenida con el modelo relajado	40
3.2. Ubicación de una caja en un subespacio maximal, generando nuevos subespacios maximales	42
3.3. Enrutamiento para la Instancia 1 (empaquetamiento factible)	45
3.4. Orden de atención para la primera ruta (vehículo 1): 3-8-7-6	46
3.5. Orden de atención para la segunda ruta (vehículo 2): 4-13-14	46
3.6. Orden de atención para la tercera ruta (vehículo 3): 9-10-15-5-12	47
3.7. Orden de atención para la cuarta ruta (vehículo 4): 11-2-1	47
4.1. Perturbación de la mejor solución encontrada hasta alguna iteración de la metodología S_{best} , para obtener S'_{best} como nuevo punto de búsqueda S_i . . .	62
4.2. Patrón de enrutamiento obtenido por el algoritmo GTS-GRASP para la instancia 1. BKS = 302.02, GTS-GRASP = 300.67	66
4.3. Patrón de empaquetamiento para la ruta: 3-8-7-6	66
4.4. Patrón de empaquetamiento para la ruta: 4-13-14	66
4.5. Patrón de empaquetamiento para la ruta: 11-9-2-1	67
4.6. Patrón de empaquetamiento para la ruta: 5-10-15-12	67

4.7. Patrón de enrutamiento obtenido por el Algoritmo GTS-GRASP para la instance 23. BKS = 1130.54, GTS-GRASP = 1122.48	68
4.8. Patrones de empaquetamiento para cada una de las rutas del BKS encontrado para la instance 23.	69
6.1. Incidencia del impacto ambiental, del 2L-HFVRP al 2L-FHFVRP.	101
6.2. Codificación y decodificación de solución. a. Conjunto de ubicaciones y piezas demandadas de los clientes. b. Ejemplo de orden de atención. c. Grafo auxiliar del ejemplo y ruta más corta. d. Conjunto de rutas y empaquetamiento dentro de estas.	103
6.3. Evolución de los espacios máximos en tres iteraciones de la fase constructiva. Adaptado de Parreno et al. (2010)	108

Índice de tablas

2.1. Estado del arte Green Vehicle Routing Problem	24
3.1. Comparación con los mejores resultados de la literatura	44
4.1. Parametrización para el Algoritmo GTS-GRASP	63
4.2. Comportamiento del algoritmo y comparación con los mejores algoritmos publicados hasta el momento del estudio	64
5.1. Parámetros utilizados en el modelo PTSP.	73
5.2. Resultados obtenidos por el Algoritmo Cut-and-Branch y por el Algoritmo ILS	89
5.3. Resultados obtenidos por el Algoritmo Genético Multi-operador (MGA) y por el Algoritmo Híbrido HA	90
6.1. Valores seleccionados para los parámetros. * Siendo n el número de clientes a rutear y m el número de cajas a ser empacadas en la ruta.	113
6.2. Resultados obtenidos por el GA-GRASP	114
6.3. Resultados obtenidos por el GA-GRASP y comparación con las mejores soluciones publicadas en la literatura (BKS)	116

Resumen

Esta investigación doctoral propone diferentes metodologías de solución para problemas de enrutamiento de vehículos considerando aspectos de carga de la vida real, permitiendo abordar adecuadamente algunas de las tareas relacionadas a la distribución de carga y logística de transporte. Dichos aspectos también tienen en cuenta el impacto ambiental, buscando minimizar la emisión de gases de efecto invernadero o el consumo de combustible.

La integración de problemas de enrutamiento de vehículos y problemas de carga de contenedores, ambos pertenecientes a la clase de problemas NP-duros, ha sido estudiada durante alrededor de una década. De hecho, se identificó que algunas variantes importantes no han sido examinadas con rigor, motivando así el desarrollo de esta investigación. De manera más precisa, se orientaron los siguientes problemas: el problema de enrutamiento-empaquetamiento con carga tridimensional (carga apilable) y el problema de enrutamiento-empaquetamiento con flota heterogénea con carga bidimensional (carga no apilable) con y sin consumo de combustible.

Abstract

This doctoral research proposes different solution methodologies for vehicle routing problems considering real-life loading aspects, which in turn may enable one to properly tackle some of the tasks related to load distribution and transportation logistics. Such aspects also take into account the environmental impact, where one intends to minimize the emission of greenhouse gases or fuel consumption.

The integration of vehicle routing and container loading problems, both belonging to the class of NP-hard problems, has been studied for just about a decade. In fact, it was identified that some important variants have not been thoroughly examined, thus motivating the development of this research. More precisely, the following problems were addressed: the routing-packing problem with three-dimensional load (stackable cargo) and the heterogeneous fleet routing-packing problem with two-dimensional load (non-stackable cargo) with and without fuel consumption.

Capítulo 1

Introducción

1.1. Planteamiento del problema

El problema de enrutamiento vehículos VRP (por sus siglas en inglés de *Vehicle Routing Problem*) es un problema clásico de la optimización combinatorial de gran interés en el campo de la investigación de operaciones debido a su influencia económica y dificultad para ser resuelto; la solución de este problema juega un papel importante en la cadena de suministros de las empresas cuyas actividades involucran transporte de bienes o personas, movimiento eficiente y efectivo de materias primas desde sus fuentes hasta las plantas de producción, productos terminados a centros de distribución y movilizaciones desde puntos minoristas hasta clientes finales. La correcta gestión de todas estas actividades es decisiva en un entorno competitivo como el que se plantea actualmente en las cadenas de abastecimiento de empresas dedicadas a la producción de bienes o a la atención de servicios.

Un mal planeamiento de las rutas de distribución puede impactar considerablemente los costos operacionales de las empresas pertenecientes a la cadena de la logística; este mal planteamiento incluso puede llegar al 50% de sobrecostos producto de inventarios en *stock* (Toth and Vigo, 2013), el incumplimiento, retrasos de entregas de pedidos a los clientes, entre otros.

En este sentido, durante las últimas 5 décadas, desde la primera formulación del VRP en 1959 por George Dantzig y John Ramser ([Dantzig and Ramser, 1959](#)), diferentes investigadores internacionales, han realizado un gran esfuerzo en el desarrollo de trabajos de investigación básica y aplicada donde se proponen nuevas formulaciones que permiten resolver diferentes variantes del problema de enrutamiento de vehículos, incorporando en cada modelo características que describen de una forma más precisa los problemas de la vida real. Dantzig y Ramser modelaron una aplicación real del problema de enrutamiento de camiones de reparto de gasolina desde un único punto de abastecimiento a un gran número de estaciones de servicio. En éste, la capacidad del camión de despacho se considera explícitamente, es decir, existen K camiones idénticos de una capacidad Q y la suma de las demandas de clientes asignados a una ruta nunca puede exceder Q , el anterior problema es denominado en la literatura especializada como CVRP (por sus siglas en inglés de *Capacitated Vehicle Routing Problem*). Este tipo de formulaciones han experimentado una evolución exitosa de la mano del desarrollo de los sistemas informáticos, con los cuales, se ha logrado un gran avance en la implementación de algoritmos cada vez más potentes, precisos y rápidos, para la resolución de un mayor espectro de problemas.

Las situaciones reales presentes en la logística, han sido las principales fuentes de motivación para el estudio de nuevas variantes del VRP, las cuales se ajustan en diferentes grados a las necesidades de los clientes, como son las ventanas de tiempo de disponibilidad de los clientes, servicios tanto de entrega como de recogida, restricciones de duración de ruta, múltiples depósitos o puntos de atención, flota heterogénea, costos fijos, entre muchos otros; sin embargo, independientemente de cuál sea la variante del problema, el objetivo principal siempre estará enmarcado en la optimización de los recursos de transporte, la satisfacción de la demanda de los clientes, y en las últimas dos décadas, ha surgido también la necesidad de disminuir de manera simultánea el impacto ambiental asociado al transporte, bien sea por tierra, por mar o por aire.

Una de las principales iniciativas de esta investigación es, enfocar los esfuerzos hacia la solución de variantes del VRP donde se consideren restricciones propias de los problemas reales de la logística de las industrias, tal como el empaquetamiento de la demanda de los clientes, que puede estar compuesta por piezas de dos dimensiones (2D) cuando se trata de elementos que no pueden ser apilables, y tres dimensiones (3D) cuando son cajas de embalaje, recogida y descarga de bienes y consideraciones de impacto ambiental del entorno donde se realiza la distribución de las mercancías; este último enfoque tiene en consideración objetivos más amplios y restricciones operativas que se ocupan de cuestiones de logística sostenible, planteando nuevos modelos de enrutamiento de vehículos y nuevos escenarios de aplicación, lo que sin duda alguna, conduce a problemas más complejos de optimización combinatoria que se ocupa de las actividades de medición de los efectos ambientales de las diferentes estrategias de enrutamiento de vehículos, reducción del consumo de energía y combustibles, administración del reciclaje y eliminación de residuos contaminantes.

1.2. Motivación y Justificación del Problema

Desplazar personas y mercancías de manera rápida, eficaz y económica es un elemento central del objetivo de una economía dinámica y una sociedad con más cohesión, las cifras de los últimos años muestran que el sector de transporte genera entre el 3 y el 10 % del Producto Interior Bruto (PIB) ([Brandao, 2009](#)). Los costos de transporte en ciudades con una densidad poblacional alta, donde la mitad de los viajes urbanos se efectúan en transporte público, en bicicleta o a pie, representan un 6 % del PIB. En contraste, las ciudades extensas alcanza hasta un 15 % del PIB, y en los países en desarrollo, donde la densidad urbana es baja, puede superar un 25 % del PIB ([Ballou, 2004](#)).

Uno de los problemas que más ha interesado a la logística urbana, ha sido el problema de enrutamiento de vehículos, por su gran número de aplicaciones dentro del diseño y administración de sistemas de distribución, como por ejemplo en el transporte de insumos y de mercancías, entregas a domicilio, rutas de autobuses públicos o escolares. Otra de las razones

por las cuales el estudio de este problema ha despertado un gran interés es su complejidad computacional a la hora de ser resuelto, ya que debido a su naturaleza, encontrar una solución óptima requiere de tiempos computacionales prohibitivos, lo cual es una de las principales razones de la existencia del número elevado de técnicas de solución implementadas hoy en día. Sin embargo, junto con la complejidad y eficiencia de dichos métodos, el problema ha involucrado cada vez más variantes que lo hacen computacionalmente más difícil de resolver, como por ejemplo, ventanas de tiempo de servicio, distribuciones de probabilidad que simulen las condiciones viales, vehículos de distintas capacidades, entre otras, dichas variantes han ido apareciendo con la necesidad de analizar y tratar de resolver problemas prácticos de la industria.

Ahora bien, de manera general el transporte de mercancías involucra dos tópicos básicos de optimización que han sido estudiados intensamente en las últimas décadas de manera aislada; las cuales consisten en encontrar las rutas óptimas para entregar los bienes, y determinar la mejor manera de cargar dichos bienes en los vehículos empleados para el transporte. La gran mayoría de los problemas que surgen en esas dos áreas pertenecen a la clase de los problemas denominados como NP-duros, y en la práctica son realmente desafiantes, lo que significa que no existe un único método de solución capaz de resolverlo en tiempo de cómputo razonable. La complejidad del VRP requiere entonces de estrategias que aporten a la eficiencia computacional.

Tradicionalmente, las decisiones de distribución de mercancía se han modelado matemáticamente usando el problema de VRP, el cual es considerado un problema de optimización combinatoria de difícil solución ([Brandao, 2009](#)), sin embargo, cuando se desean abordar situaciones prácticas, el VRP por si sólo no representa en detalle el problema real que se desea solucionar, por lo que es preciso, mejorar el modelo idealizado por medio de la incorporación de un conjunto de restricciones y características adicionales tales como, número de depósitos de abastecimiento, ventanas de tiempo, capacidad de los vehículos, demanda de los clientes, ubicación de los clientes, tamaño de la flota de vehículos que puede ser homogénea o heterogénea, entregas y recogidas en la misma ruta, componentes estocásticos y dinámicos, rutas periódicas, pedidos mínimos, duración total del recorrido, entre muchas otras, que

incrementan la complejidad de los modelos matemáticos e incrementa el nivel de dificultad a la hora de diseñar métodos solución apropiados.

Es evidente que ajustar los diferentes modelos matemáticos del problema clásico de VRP para aplicaciones reales de la industria, requiere la adición de nuevas restricciones técnicas propias del problema a resolver. La variante denominada enrutamiento-empaquetamiento, en la cual, las demandas se componen de ítems discretos, tales como: cajas rectangulares, equipos de limpieza, *pallets* y otros vehículos de menor tamaño. Esta consideración, introduce restricciones de empaquetamiento que no pueden ser descritas como un problema de una sola dimensión, donde se asuma implícitamente que cada demanda ocupa una determinada sección del objeto y este se ajusta a la forma del contenedor, es decir, como si se tratase de un líquido, sino que por el contrario, requiere de ciertas características espaciales y dimensionales. La representación realista de este aspecto del problema, ha despertado el interés por nuevas variantes del problema que combina enrutamiento y empaquetamiento; denominadas en la literatura especializada como 2L-CVRP, 2L-HFVRP, 3L-CVRP y 3L-HFVRP.

Consecuentemente, los problemas de transporte que involucran enrutamiento y carga de manera simultánea, son actualmente de gran interés en optimización combinatorial, lo cual radica en la dificultad intrínseca de esta área de investigación que combina dos problemas computacionales de difícil solución con una gran relevancia práctica en aplicaciones reales, además, considera la ubicación factible de los ítems demandados en el espacio de carga, donde se presentan restricciones relevantes y frecuentes, que consisten en que la ubicación de las cargas permita realizar las operaciones de entrega sin redistribuir los ítems dentro del contenedor en que se están transportando, lo que evita el desperdicio de tiempo cuando el vehículo visita un cliente (menor tiempo de servicio).

Es evidente entonces, que los servicios de transporte en general facilitan el crecimiento económico y social de las ciudades, sin embargo, son los medios de transporte los que generan mayores impactos ambientales negativos, como la utilización de terrenos, Emisiones de Gas de Efecto Invernadero GHG (del inglés, *Greenhouse Gas*), polución, ruido, niebla de verano y efectos tóxicos en el ecosistema como la lluvia ácida, lo cual se evidencia en el estudio

publicado por la Agencia Ambiental Europea donde se establece que el transporte (incluyendo transporte internacional marítimo) aportó el 24 % del total de emisiones GHG en los países del (EU-27-European-Union, 2013) en el año 2013, teniendo el transporte de carreteras una cuota del 17 % del total de las emisiones GHG (Vicente, 2011)

Ante este panorama, las diferentes políticas públicas de cada país han involucrado un elemento adicional en el quehacer diario de las empresas, las cuales deben integrar a sus planes estratégicos, la responsabilidad ambiental en el desempeño de sus actividades; especialmente, el sector transporte, responsable de producir el mayor porcentaje de emisiones generadas a partir de combustibles fósiles cercanas al 30 % (US-EPA), evidenciándose en recientes investigaciones que el CO_2 es el principal causante de Emisiones de Gases de Efecto Invernadero (GHG) en la actividad del transporte urbano de mercancías, y es emitido en proporción directa al consumo y tipo de combustible (Grant et al., 2008). Para la mayoría de vehículos, el consumo de combustible y la tasa de emisión de CO_2 por kilómetro recorrido disminuye cuando la velocidad operativa del vehículo aumenta entre 80 y 100 km/h aproximadamente, y luego comienza a incrementar de nuevo para velocidades superiores (Grant et al., 2008), esto indica una relación no lineal entre la tasa de emisiones y la velocidad del recorrido del vehículo.

Ante la situación descrita anteriormente, se propuso un trabajo de investigación donde se busca optimizar el plan de entrega de productos a un conjunto de clientes demandantes, cuyo diseño de rutas considere la carga de los bienes a los vehículos, la fragilidad de los mismos, la estabilidad de carga y la descarga de los bienes de los vehículos al visitar los clientes, lo anterior, repercute en restricciones de empaquetamiento bidimensional y tridimensional que surgen para la distribución de las demandas en la flota y representan una mayor dificultad para su verificación y agregan una complejidad alta a los mecanismos de solución, dado que se resuelven de manera simultánea dos problemas NP-duros que conforman uno solo enrutamiento-empaquetamiento (Gendreau et al., 2006), que pueden ser representados y aplicados en la industria de la cadena de abastecimiento de madera, cadena de distribución de supermercados, distribución de paquetes postales, entre otras; estos y muchos otros ejemplos requieren de planeación de rutas que tengan en cuenta detalles reales para ubicar la carga,

prevenir su deterioro mientras se transporta, se acomoda o se extrae del contenedor.

Cabe destacar que los problemas abordados en esta propuesta, son variantes del VRP que han sido poco explorados, como es el caso del problema que considera restricciones de dos dimensiones en el empaquetamiento (carga no apilable) y flota heterogénea en el enrutamiento el que se denomina preciso de 2L-HFVRP (del inglés *Two-Dimensional Container Loading Problem With Heterogeneous Fleet Vehicle Routing Problem*), para lo cual se propone como método de solución, la implementación de una metaheurística híbrida, basada en dos algoritmos poblacionales, un segundo problema a abordar es la variable denominada 3L-CVRP (del inglés *Three-Dimensional Container Loading Problem With Capacitated Vehicle Routing Problem*), el cual considera restricciones tridimensionales en el empaquetamiento (carga apilable, fragilidad de la carga y rotación) y capacidad de la flota en el enrutamiento, para este caso se propone desarrollar un método de solución matheurístico, que combina el algoritmo de *Branch-and-Cut* para solucionar el problema de enrutamiento y un algoritmo GRASP para el empaquetamiento, para lo cual se proponen modificaciones al modelo tradicional de CVRP de dos índices.

Por otro lado, esta investigación propone el planteamiento de un nuevo modelo del problema de *Pollution Routing Problem*, el cual no ha sido estudiado actualmente, para lo cual se propone un nuevo modelo matemático y un método de solución especializado exacto, que combina el método de *Cut and Branch* con el algoritmo de separación diseñado por ([Padberg and Rinaldi, 1990](#)).

Finalmente, en este trabajo se integran el empaquetamiento, el enrutamiento, y el impacto ambiental, en una nueva variante de esta familia de problemas, denominada 2L-FHVRP (del inglés *Two-Dimensional Container Loading Problem With Heterogeneous Fleet Vehicle Routing Problem Considering Fuel Consumption*), y será abordada en el capítulo final de este documento.

1.3. Objetivos

1.3.1. Objetivo General

Plantear modelos y técnicas de solución adecuadas para problemas de enrutamiento-empaquetamiento considerando restricciones multidrop e impacto ambiental.

1.3.2. Objetivos Específicos

- Identificar frentes de investigación para los problemas de enrutamiento-empaquetamiento considerando restricciones multidrop.
- Proponer metodologías de solución adecuadas para el problema 2L-HFVRP.
- Proponer metodologías de solución adecuadas para el problema 3L-CVRP.
- Validar metodologías de solución propuestas con casos de la literatura especializada para las variantes CLP-VRP abordadas.
- Proponer metodologías de solución para problemas de enrutamiento que consideran impacto ambiental.
- Proponer metodologías de solución para el problema 2L-FHFVRP.

1.4. Estructura del documento

El documento está organizado de la siguiente manera: el Capítulo 2, ofrece el marco referencial y teórico para los subproblemas y problemas tratados en esta investigación. Luego, en los Capítulos 3 y 4 se aborda el primer problema combinado de subproblemas NP-Duros (enrutamiento y empaquetamiento) sin consideraciones ambientales. Posteriormente, en el Capítulo 5, se presenta un estudio dedicado a problemas de enrutamiento de vehículos con impacto ambiental, y se propone un problema nuevo en este frente, junto con dos propuestas de solución diferentes. Finalmente, en el Capítulo 6, se aborda una variante no estudiada

de problemas de enrutamiento y empaquetamiento combinados, considerando su impacto ambiental por medio de la minimización del consumo de combustible.

Al tratarse de un estudio de variedad de problemas de enrutamiento-empaquetamiento, cada capítulo presenta su estudio computacional y conclusiones correspondientes, sin embargo, en el Capítulo 7 se presentan las conclusiones generales de todo el proceso, junto con trabajos futuros relevantes que aportarían directamente sobre el estado del arte de este estudio. En este mismo capítulo se presenta la producción más importante obtenida durante el ejercicio de esta investigación.

Capítulo 2

Marco Referencial

En este capítulo se presentan los marcos referenciales de las variantes de enrutamiento estudiadas en este trabajo, con el fin de contextualizar los estudios comprendidos en esta investigación.

2.1. Marco referencial enrutamiento de vehículos y carga de contenedores

2.1.1. Enrutamiento de vehículos

El problema de enrutamiento de vehículos, se ha estudiado por más de 50 años ([Laporte, 2009](#)) y se ha resuelto usando técnicas exactas y técnicas aproximadas tales como, las heurísticas y metaheurísticas, sin embargo, durante los últimos cinco años se pueden encontrar soluciones híbridas entre diferentes técnicas que han mostrado resultados de muy buena calidad (([Brandao, 2009](#)), ([Prins, 2009](#)), ([Subramanian et al., 2011](#)), ([Liu et al., 2009](#))), así como, investigaciones direccionadas en nuevas variantes de problema clásico, que se ajustan en mayor medida a la realidad de los problemas de las industrias, tal como los que se describen a continuación.

Los problemas de transporte involucrando enrutamiento y carga de manera simultánea, son actualmente de gran interés en optimización combinatorial. El interés de los investigadores y profesionales está motivado por la dificultad intrínseca de esta área de investigación que combina dos problemas computacionales difíciles, y por su relevancia práctica en aplicaciones reales.

Las raíces de los problemas de enrutamiento datan del siglo XIX, cuando el matemático irlandés William Rowan Hamilton, definió el Problema del Circuito Hamiltoniano: decidir si existe una secuencia consecutiva de arcos del grafo visitando cada uno de los vértices una sola vez. Su extensión al caso con pesos dio origen al famoso problema del Agente Viajero TSP (del inglés *Traveling Salesman Problem*) definido en la década de los treinta por Karl Manger, problema sobre el que se iniciaron estudios intensivos en la década de los cincuenta (con Dantzig y Fulkerson, entre otros). Dado un grafo $G = (V, E)$, con un conjunto de vértices $V = \{0, 1, \dots, n\}$, conjunto de aristas $E = \{(i, j) : i, j \in V, i \neq j\}$ y costo c_{ij} para viajar a través de la arista (i, j) (en cualquier dirección), el problema del Agente Viajero Simétrico STSP (del inglés *Symmetric Traveling Salesman Problem*) consiste en encontrar una secuencia de aristas consecutivas (circuito) que visite todos los vértices del conjunto V a un costo mínimo total. Su contraparte asimétrica ATSP (del inglés *Asymmetric Traveling Salesman Problem*) es modelada de manera similar en un dígrafo $G = (V, A)$ teniendo un conjunto de arcos $A = \{(i, j) : i, j \in V, i \neq j\}$ y c_{ij} como el costo de viajar desde i hasta j . La literatura para el TSP incluye cientos de publicaciones, recopilaciones y libros. En (D'Ambrosio et al., 2011) y (Letchford and Lodi, 2011) son presentadas dos publicaciones enciclopédicas que tratan ampliamente este problema clásico de investigación de operaciones.

En los problemas de transporte reales, el TSP es generalizado como Problema Capacitado de Enrutamiento de Vehículos, en el cual se busca, en lugar de un solo circuito, un conjunto de circuitos (llamados rutas) iniciando y terminando en el depósito central ubicado en el vértice 0. Una formulación completa del problema presentada en (Toth and Vigo, 2013), comúnmente conocida como formulación de dos índices para el problema de enrutamiento de vehículos capacitado es la siguiente:

$$\text{Min} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.1)$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \forall j \in V \setminus \{0\} = 1 \quad (2.2)$$

$$\sum_{j \in V} x_{ij} = 1, \forall i \in V \setminus \{0\} = 1 \quad (2.3)$$

$$\sum_{j \in V} x_{0j} = k \quad (2.4)$$

$$\sum_{i \in V} x_{i0} = k \quad (2.5)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S), r(S) = \left\lceil \frac{d(S)}{C} \right\rceil \quad (2.6)$$

Los conjuntos de restricciones del modelo controlan los siguientes aspectos:

- Las ecuaciones (2.2) y (2.3) representan los conjuntos de restricciones para controlar el grado de entrada y de salida a los puntos donde se encuentran ubicados los clientes. En otras palabras, se garantiza que a cada cliente llegue un solo vehículo e igualmente salga un solo vehículo.
- Los conjuntos (2.4) y (2.5) controlan que el mismo número de rutas salgan y entren del depósito (nodo 0).
- El conjunto de restricciones (2.6) valida a través de la diferencia entre la cardinalidad de cualquiera de los posibles subconjuntos de clientes S y el número requerido de vehículos para atender la demanda de los clientes pertenecientes a este subconjunto denotada por $r(S) = \left\lceil \frac{d(S)}{C} \right\rceil$, donde $d(S)$ es la demanda total del subconjunto S de clientes y C es la capacidad de los vehículos. En este caso C es igual al volumen de cada vehículo. Con esta diferencia, se garantiza que no se formen subtours y que no se exceda la demanda en las rutas de manera simultánea.

Debido a la explosión exponencial de subconjuntos posibles de clientes que son controlados por la ecuación (2.6), la solución de este modelo se dificulta cuando el número de nodos se incrementa. La alta complejidad de este problema ha favorecido a que tanto áreas de la optimización clásica como aproximada se interesen en la resolución de este. El problema ha sido investigado profundamente desde la década de los cincuenta. Compilaciones de gran interés que abordan el tema son presentadas por [Toth and Vigo \(2013\)](#), [Golden et al. \(2008\)](#) y [Baldacci et al. \(2010\)](#). Mientras que [Schmid et al. \(2013\)](#) hace una compilación extendida sobre una variante del problema, en la cual la capacidad de la flota de vehículos es heterogénea.

A continuación una revisión de la literatura relevante sobre el problema de empaquetamiento tridimensional.

2.1.2. Carga de contenedores

En problemas reales de empaquetamiento, la carga de los clientes no solo está caracterizada por un valor (como en el caso del CVRP), sino también, por su forma y ubicación en el espacio. Es necesario asegurarse de que los ítems transportados puedan ser ubicados de manera factible dentro del espacio de carga que ofrece el vehículo, estas especificaciones de empaquetado se relacionan con los problemas de empaquetamiento rectangular multidimensional, los cuales surgen como una extensión del problema clásico de bin-packing unidimensional, el cual puede ser descrito como el problema de ubicación, sin traslape, de un conjunto de segmentos, cada uno teniendo un ancho, para un número mínimo de segmentos grandes idénticos. Las principales extensiones para dimensiones superiores son:

- **Problema Bin Packing Bidimensional (2BPP):** Empacar un conjunto de rectángulos dentro de un número mínimo de rectángulos grandes (contenedores) idénticos (bins).
- **Problema Strip Packing Bidimensional (2SPP):** Empacar un conjunto de rectángulos en una tira (strip) con final abierto y ancho dado, con una altura infinita

buscando minimizar la altura total empleada.

- **Problema Bin Packing Tridimensional (3BPP):** Empacar un conjunto de cajas rectangulares dentro de un número mínimo de contenedores idénticos mayores a las cajas.
- **Problema Strip Packing Tridimensional (3SPP):** Empacar un conjunto de cajas rectangulares dentro de una tira tridimensional con final abierto con ancho y alto dados y una longitud infinita, buscando minimizar la longitud total de uso de la tira.
- **Problema Knapsack Tridimensional (3SKP):** Empacar de un conjunto de cajas rectangulares dentro de un contenedor buscando maximizar el volumen ocupado por la carga empacada.

De manera similar al problema clásico de VRP, el problema de empaquetamiento en sus diferente variantes ha sido estudiado ampliamente, una introducción general al área de empaquetamiento rectangular es presentada por (Lodi et al., 2002), (Lodi et al., 2010), (Wascher et al., 2007) y (E.G. Coffman et al., 2013).

En esta tesis se resuelve el problema integrado. En la literatura, los primeros acercamientos para solucionar el problema de forma integrada consideran la relajación del problema de empaquetamiento tridimensional al considerar la carga como elementos bidimensionales.

2.1.3. Problema integrado de enrutamiento de vehículos y carga de contenedores

En este trabajo se aborda el problema de empaquetamiento descrito, junto con el problema de enrutamiento de vehículos. En la literatura, los primeros acercamientos para solucionar el problema de forma integrada, efectuaron la relajación del problema de empaquetamiento tridimensional al considerar la carga como elementos bidimensionales y se han empleado diferentes metodologías de solución, la mayoría de estas presentan algoritmos aproximados debido a la complejidad de generar un procedimiento exacto eficiente para resolver este problema integrado.

En (Iori, 2005) se presenta un resumen general de algoritmos heurísticos para el 2L-CVRP y el primer algoritmo exacto para la variante secuencial 2L-CVRP con aristas de costo entero es presentado en (Iori et al., 2007), en donde se manejan dos restricciones adicionales que en ocasiones aparecen en la literatura del CVRP: (i) Todos los K vehículos deben ser utilizados, y (ii) no se permiten rutas de un solo cliente. El algoritmo se basa en una aproximación *Branch-and-Cut* que utiliza desigualdades válidas para remover secuencias de empaquetamiento infactibles, la factibilidad del patrón de carga es evaluada a través de heurísticas y procedimientos *Branch-and-Bound* anidados.

Luego, en (Gendreau et al., 2007) se presenta el primer algoritmo metaheurístico, basado en búsqueda tabú para el 2L-CVRP, en esta metodología, se maneja tanto la versión de empaquetamiento secuencial como la irrestricta; el algoritmo puede aceptar movimientos que producen rutas infactibles tanto por exceso de peso como por patrones de empaquetamiento que excedan la altura de la superficie de carga.

En (Fuellerer et al., 2009), los autores presentan un algoritmo de optimización de colonia de hormigas ACO (del inglés *Ant Colony Optimization*) en donde cada hormiga busca una solución factible de bajo costo a través de la generalización del algoritmo clásico de ahorros, la factibilidad de carga es revisada a través de límites inferiores (*lower bounds*), heurísticas y un algoritmo de *Branch-and-Bound* truncado.

Un algoritmo metaheurístico de Búsqueda Tabú es presentado en (Zachariadis et al., 2009), el cual utiliza un mecanismo de guía que dirige la búsqueda hacia áreas de solución altamente diversas, la información en las rutas generadas es almacenada en tablas *hash* para evitar ejecuciones redundantes.

De manera similar, un algoritmo de búsqueda de vecindario variable VNS (del inglés *Variable Neighborhood Search*) en el que un intercambio de parámetros de vecindario es utilizado para perturbar la solución sobre la que se está iterando y una búsqueda local de 2-opt convencional determina un óptimo local, es presentado en (Strodl et al., 2010).

Dos algoritmos metaheurísticos presentados en (Leung et al., 2010) y (Leung et al., 2011) utilizan una heurística basada en ajuste para producir patrones de empaquetamiento factibles.

El primero explora la solución espacial a través de tres vecindarios diferentes, mientras que el segundo está basado en una búsqueda local guiada.

En (Duhamel et al., 2011) el autor presenta una heurística que comienza con un algoritmo de procedimiento de búsqueda golosa adaptativa y aleatorizada GRASP (del inglés *Greedy Randomized Adaptive Search Procedure*) que genera una solución de un solo tour y la divide en rutas separadas. Las rutas resultantes son optimizadas a través de una combinación de herramientas genéticas (mutaciones) y búsquedas locales, finalmente, un algoritmo genético para el problema 2L-CVRP es presentado en (Shen and Murata, 2012.), este trabajo no ofrece comparaciones con algoritmos previos.

Problema de enrutamiento de vehículos capacitado con carga bidimensional y flota heterogénea (2L-HFVRP)

La variante 2L-HFVRP es uno de los problemas que presenta gran interés en aplicaciones prácticas, sin embargo, ha sido poco explorado en la literatura especializada, el primero de los trabajos publicados, se presenta en (Leung et al., 2013), los autores consideran diferentes tipos de vehículos, con diferente capacidad en las dimensiones de ancho y largo. De acuerdo a la aplicación específica, el costo asociado al uso del vehículo puede ser fijo o variable, el método de solución propuesto corresponde a un algoritmo de recocido simulado que emplea una heurística de búsqueda local para mejorar la solución encontrada, además un número de algoritmos heurísticos de empaquetamiento son utilizados para manejar las restricciones de carga.

Finalmente, el mismo autor presenta en (Dominguez et al., 2016) un estudio de las características del problema, y propone un algoritmo multi-arranque basado en aleatorización parcial de heurísticas para enrutamiento y empaquetamiento, y una continuación del trabajo anterior, es presentada en (Rivero et al., 2016), donde utiliza un framework ILS (del inglés *Iterated Local Search*) parcialmente aleatorizado para heurísticas de enrutamiento y empaquetamiento.

Problema de enrutamiento de vehículos capacitado con carga tridimensional y flota homogénea 3L-CVRP

Otra de las variantes del problema de enrutamiento - empaquetamiento, es presentada en ([Gendreau et al., 2006](#)), donde se formula el problema conocido como 3L-CVRP y se presenta un algoritmo que se encuentra basado en un marco de trabajo externo de búsqueda Tabú, por cada solución vecina generada, las cargas de los vehículos son obtenidas invocando una búsqueda Tabú interna en un 3SPP (del inglés *Three-Dimensional Strip Packing Problem*), si la carga resultante excede la longitud del vehículo, la solución retornada a la búsqueda externa es aceptada, pero con su penalización correspondiente. El algoritmo es probado tanto en instancias generadas aleatoriamente como en instancias reales de 3L-CVRP ofrecidas por una compañía de muebles.

En ([Tarantilis et al., 2009](#)), los autores proponen una adaptación al caso tridimensional de su búsqueda Tabú guiada para el caso bidimensional y en ([Fuellerer et al., 2010](#)) presenta una generalización al caso tridimensional del ACO de los autores para la variante bidimensional presentada previamente.

El núcleo del enfoque presentado en ([Tao and Wang, 2010](#)), es una heurística de mínimo desperdicio para la solución del subproblema de carga iterativamente invocado por un algoritmo de búsqueda Tabú externo.

Aproximaciones heurísticas en las cuales el subproblema de enrutamiento es resuelto por un algoritmo metaheurístico (apareamiento de abejas para el primero, algoritmos genéticos en el segundo) son presentadas en ([Ruan et al., 2013](#)) y ([Miao et al., 2012](#)), en estas, el subproblema de carga es resuelto a través de una búsqueda Tabú o de heurísticas constructivas, respectivamente, los experimentos computacionales muestran que la metodología presentada en ([Miao et al., 2012](#)) es más eficiente.

En ([Zhu et al., 2012](#)) dos heurísticas de empaquetamiento de la literatura son mejoradas y embebidas en un algoritmo de búsqueda Tabú. Experimentos computacionales muestran la eficiencia del enfoque propuesto, así como, un modelo integrado multi-día para el problema

de transporte tridimensional de fletes es considerado en ([Attanasio et al., 2007](#)), los autores proponen un modelo de Programación Lineal Entera ILP (del inglés *Integer Lineal Problem*), adoptando una técnica *rollinghorizon*, y un algoritmo heurístico para las restricciones de empaquetamiento; el método es validado con datos reales provenientes de una compañía de químicos.

Un problema de enrutamiento de vehículos con restricciones de carga tridimensional, pero sin restricciones de capacidad es presentado en ([Koloch and Kaminski, 2010](#)), en el cual, se proponen dos enfoques cuyos resultados son comparados entre si, sin embargo, los autores no mencionan resultados previos de la combinación del enrutamiento y del empaquetamiento.

En ([Bortfeldt and Homberger, 2013](#)), se desarrolla una heurística de dos etapas para el problema considerado por ([Moura and Oliveira, 2009](#)), en la primera etapa se optimiza el empaquetamiento, mientras que la segunda, se encarga del aspecto de las rutas. Los experimentos computacionales muestran la alta eficiencia del método.

En el trabajo realizado en ([Bortfeldt, 2012](#)), el autor presenta una aproximación híbrida eficiente, basada en un algoritmo de búsqueda Tabú para el subproblema de enrutamiento, en cada una de las iteraciones de la búsqueda Tabú, las rutas generadas son ordenadas en una lista, la cual está ordenada de manera creciente según el costo de la ruta y por cada solución en la secuencia resultante, un algoritmo de búsqueda en árbol soluciona el subproblema de carga. Los experimentos computacionales muestran la efectividad de la metodología propuesta.

Todos los trabajos presentados anteriormente, proponen como método de solución una estrategia heurística o metaheurística, siendo el único trabajo que emplea un método exacto de solución en su estructura, la investigación presentada en ([Escobar-Falcón et al., 2016](#)). Un procedimiento basado en cortes obtiene soluciones del problema de enrutamiento que luego son validadas a través de un GRASP, el cual revisa las restricciones de empaquetamiento de cada una de las rutas encontradas o sugeridas por la solución del modelo creciente. Para el algoritmo híbrido se utiliza la relajación del modelo clásico de CVRP de dos subíndices ([Toth and Vigo, 2013](#)), y diferentes tipos de cortes son adicionados: eliminación de subtours, cortes debido a las restricciones de capacidad y cortes para restricciones de empaquetamiento.

Otras variantes de enrutamiento-empaquetamiento

Para un número relevante de variantes clásicas del CVRP (ventanas de tiempo, flota heterogénea, etc.), la adición de variantes de restricciones de carga ha sido considerada en la literatura reciente.

En (Moura and Oliveira, 2009) se aborda la dificultad de integrar el CVRP con ventanas de tiempo y la carga de contenedores tridimensional, proponiendo un número de heurísticas constructivas. El siguiente estudio (Moura, 2008), extiende los resultados a un caso multiobjetivo, cuya función objetivo considera el número de vehículos, la distancia total viajada, y el volumen utilizado y para resolver el problema se presenta la implementación de un algoritmo genético.

En la variante considerada en (Khebbache-Hadji et al., 2013), cada cliente i ($i = 1, 2, \dots, n$) tiene una ventana de tiempo $[a_i, b_i]$ el vehículo asignado al cliente debe entregar los bienes ni más temprano que a_i ni más tarde que b_i . Este trabajo provee algunas heurísticas constructivas y un algoritmo genético.

Una nueva restricción del problema son introducidas en (Hamdi-Dhaoui et al., 2012), en donde algunas parejas de ítems no pueden ser transportados por el mismo vehículo, lo cual es típico en el transporte de materiales peligrosos. Los autores utilizan un algoritmo genético para resolver esta variante del problema.

La variante principal considerada en (Martínez and Amaya, 2013) consiste en que los ítems transportados presentan forma circular. Los autores presentan una heurística constructiva y una búsqueda Tabú para mejorar la calidad de la solución, siendo una aplicación real de ítems circulares el transporte de paellas.

2.2. Marco referencial Green Vehicle Routing

El daño ambiental causado por las actividades de transporte ha sido reconocido desde la década del 50 (McKinnon, 2010), aunque no se realizó un trabajo realmente significativo hasta

comienzos del 2000, los impactos ambientales más destacados en el transporte de mercancía son los siguientes:

- *Emisiones atmosféricas:* Estas se deben a la combustión de motores utilizados en vehículos para el transporte de carga, los cuales, al convertir el combustible en energía, emiten contaminantes como CO , CO_2 , NO_X y partículas en suspensión, dichos contaminantes, tienen efectos nocivos en los humanos (problemas respiratorios y asma) y en el medio ambiente (lluvia ácida y niebla de verano) (Cullinane and Edwards, 2010)
- *Noise pollution:* Las tres fuentes generadoras de ruido son la propulsión, el contacto entre ruedas/carretera y la aceleración (Cullinane and Edwards, 2010), el ruido generado por la unidad de poder de un vehículo, comprendido por el motor, la toma de aire y el exosto se vuelven dominantes a velocidades bajas entre 15 a 20 mph y en tazas de alta aceleración de $2 \text{ m} / \text{s}^2$ (Knight, 2004), destacando molestia, dificultades de comunicación y pérdida de sueño como consecuencias en la vida humana, aunque también, la vibración de la carretera causada por vehículos muy pesados puede dañar las edificaciones próximas a la misma con el paso del tiempo.

En (McKinnon, 2010) son identificados nueve factores que son críticos para la reducción de externalidades de las actividades de logística, estas son: número de pasajeros por vehículo, tonelaje o peso de la carga transportada, utilización del vehículo (promedio del vehículo funcionando vacío y con carga), eficiencia energética, emisiones y otras externalidades que no pueden ser medidas a través del consumo energético (ruido y accidentes), y la valoración monetaria de las externalidades. De manera general, es difícil calcular los costos causados por las externalidades asociadas al transporte, existen estimaciones para camiones inter-ciudad de transporte de carga, tal como se se presenta en (Forkenbrock, 1999), donde se proponen cuatro tipos generales de costos: accidentes, emisiones, ruido y aquellos asociados al suministro, operación y mantenimiento de instalaciones públicas. En (Forkenbrock, 2001) se presenta un análisis similar para trenes de carga y se comparan sus costos externos con los asociados al transporte de mercancía con camiones, donde la conclusión general de dicha comparación, es que las externalidades asociadas al transporte con camiones es de más de tres veces las

generadas por trenes de carga.

De los diferentes tipos de externalidades mencionadas, una revisión de la literatura emergente de *Green Vehicle Routing* muestra una creciente tendencia estudiando las emisiones y el consumo de combustible, específicamente, la minimización en la planeación de rutas de operación (Eglese and Black, 2012). Las consecuencias perjudiciales de las emisiones de GHG, subproducto generado por la utilización de combustibles fósiles, y su relevancia en las economías, lo convierten en un foco de investigación lógico para aminorar el impacto ambiental de las actividades de transporte.

2.2.1. Consumo de combustible y modelos de emisión para transporte de bienes por carretera

La cantidad de CO_2 emitido por un vehículo es directamente proporcional al consumo de combustible. En la literatura se sugieren dos maneras de estimar el consumo de combustible de los vehículos: *medidas en carretera*, que se basa en la recolección en tiempo real de los datos correspondientes a las emisiones producidas por el vehículo funcionando, y el *modelo analítico de consumo de combustible o emisiones*, los cuales estiman el consumo de combustible basados en una serie de parámetros del vehículo, del medio ambiente y concernientes al tráfico de las vías como la velocidad del vehículo, carga y aceleración del mismo. En (Ardekani et al., 1992), (Esteves-Booth et al., 2002) y (Boulter et al., 2007) se encuentran revisiones detalladas de modelos de emisiones producidas por vehículos.

Los modelos analíticos para el consumo de combustible pueden ser clasificados de manera muy general en tres clases: (i) modelos con factor de emisiones, (ii) modelos basados en velocidad promedio y (iii) modelos instantáneos (Esteves-Booth et al., 2002). Los modelos con factor de emisiones son los más simples en forma y son utilizados a gran escala (estimaciones de emisiones a nivel regional o nacional), particularmente cuando los datos relacionados a las jornadas de los vehículos son limitados. Estos modelos normalmente utilizan un factor de emisiones frecuentemente expresados por unidad de distancia. Los modelos basados en velocidad promedio utilizan funciones que relacionan velocidad para estimar emisiones al

nivel de redes de carreteras y no incluyen parámetros detallados para realizar un análisis microescala. Finalmente los modelos modales o instantáneos operan a un nivel de complejidad más alto, especificidad suficiente para ser utilizados a un nivel microescala y utilizan entradas detalladas como aceleración, gradiente de la carretera, los cuales son obtenidos del motor de vehículo en funcionamiento incluso con una toma segundo a segundo de esta información. Dado que los modelos basados en factor de emisiones son bastante simplistas y presentan desventajas por su incapacidad para representar ciclos de conducción con buena exactitud, centraremos nuestra atención principalmente en los modelos (ii) y (iii). Para más detalles de los modelos con factor de emisiones ([Esteves-Booth et al., 2002](#)).

Modelos basados en velocidad promedio

A continuación se describen brevemente dos modelos para la estimación de emisiones basados principalmente en velocidad y se obtienen utilizando técnicas de regresión. El primero de esos se debe al reporte MEET publicado por la Comisión Europea ([Hickman et al., 1999](#)), donde los autores presentan la siguiente expresión general para calcular la tasa de emisiones $E(v)$ (g/km) para un vehículo vacío de transporte de carga en una carretera de gradiente cero en función de la velocidad promedio v del vehículo (km/h):

$$E(v) = \zeta_0 + \zeta_1 v + \zeta_2 v^2 + \zeta_3 v^3 + \zeta_4/v + \zeta_5/v^2 + \zeta_6/v^3 \quad (2.7)$$

Donde $\zeta_0 - \zeta_6$ son coeficientes predefinidos para diferentes tipos de vehículos clasificados por su peso o tonelaje. El reporte MEET describe funciones similares para corrección de factores para aspectos adicionales, como el gradiente y la carga, para ser aplicados a $E(v)$. Por ejemplo la tasa de emisiones de acuerdo al reporte MEET para un vehículo de menos de 3.5 toneladas de peso está dado por $E(v) = 0,0617v^2 - 7,8227v + 429,51$. Otro modelo o aproximación relevante basado en velocidad promedio denominado COPERT es descrito en ([Ntziachristos et al., 2000](#)).

Modelos instantáneos

Los modelos de esta categoría manejan solamente emisiones en caliente; es decir, emisiones exhaustivas del motor del vehículo en funcionamiento, y apuntan a la estimación de las tasas de emisión de un vehículo operando durante intervalos cortos de tiempo de su ciclo de conducción, con una frecuencia por ejemplo de segundo a segundo. Dado que estos modelos requieren medidas detalladas y precisas del vehículo en operación, las cuales suelen ser difíciles y costosas de recolectar, estos modelos se recomienda sean utilizados para investigación solamente ([Boulter et al., 2007](#)).

Minimizando Emisiones en VRP Las primeras investigaciones que contabilizan las emisiones o el consumo de combustible en el VRP inicialmente se concentraron en el peso del vehículo vacío como el factor determinante de las emisiones y por ende el elemento principal a optimizar. Trabajos posteriores se centraron en la optimización de la velocidad, combinada también con el peso del vehículo y algunos trabajos incluyeron otros factores adicionales. Algunos trabajos analizaron el efecto de la dependencia del tiempo de viaje.

La **independencia del tiempo** en el contexto del VRP implica la suposición de que los datos del problema, en particular los tiempos de viaje entre los nodos o clientes, no cambia con el tiempo.

La **dependencia del tiempo** por el contrario, permite que los datos del problema cambien con el tiempo. Los cambios en el tiempo de viaje a través de una sección de la carretera pueden ser atribuidos a eventos esperados como la congestión producida en la hora pico, o eventos inesperados como accidentes de tránsito o cambios bruscos en el clima.

En este estudio nos centraremos en modelos donde no hay dependencia del tiempo, pero las velocidades son variables y por tanto son variables de decisión del problema. Este enfoque en particular es apropiado para este estudio porque tiene un alcance suficiente de análisis al asumir las velocidades de manera variable, sin dificultar el desarrollo del estudio estableciendo dependencia del tiempo que requiere un insumo de datos costosos y difíciles de obtener. En la Tabla [2.1](#) se presenta el estado del arte de la variante de interés, contextualizada con el

Tabla 2.1: Estado del arte Green Vehicle Routing Problem

	<i>Independientes del Tiempo</i>	<i>Dependientes del Tiempo</i>
<i>Velocidad Fija</i>	Kara et al. (2007)	Kuo (2010)
	Palmer (2007)	Eglese et al. (2006)
	Ruzinski et al. (2008)	Maden et al. (2010)
	Suzuki (2011)	Conrad and Figliozzi (2010)
	Ubeda et al. (2011)	Figliozzi (2010)
	Xiao et al. (2012)	Figliozzi (2011)
<i>Velocidad Variable</i>	Bektaş and Laporte (2011)	Franceschetti et al. (2013)
	Demir et al. (2012)	Jabali et al. (2012)
	Kramer et al. (2015)	Qian and Eglese (2016)

estado del arte de las variantes que pertenecen al Green VRP.

2.3. Marco teórico

2.3.1. Definición del problema integrado de enrutamiento de vehículos y carga de contenedores

Carga bidimensional en los vehículos

En este tipo de problema tradicionalmente se parte del CVRP, en donde, para la demanda de un cliente i ($i = 1, 2, \dots, n$), el peso total d_i es determinado por m_i ítems. Cada ítem I_{il} ($l = 1, 2, \dots, m$) tiene ancho w_{il} y alto h_{il} , mientras la superficie de carga de cada vehículo tiene ancho W y alto H . Sea $S(k) \subseteq \{1, 2, \dots, n\}$ el conjunto de clientes visitados por el vehículo k . Adicionalmente a la restricción de capacidad clásica del CVRP (2.6), una solución requiere una carga factible, que para este caso consiste en un patrón de ubicación donde no exista traslape entre ninguno de los ítems rectangulares requeridos por los clientes pertenecientes a $S(k)$ dentro del área de carga $W \times H$.

El problema resultante, denominado Problema de Ruteo de Vehículos Capacitado con Restricciones de Carga Bidimensional y denotado 2L-CVRP (del inglés *Two-Dimensional Container Loading Problem With Capacitated Vehicle Routing Problem* Problema de Ruteo de Vehículos Capacitado con Restricciones de Carga Bidimensional) sea por su fragilidad o por su peso.

Un número de variantes de 2L-CVRP ha sido considerado en la literatura, básicamente con dos criterios adicionales:

- *Orientación:* Los ítems pueden tener orientaciones fijas, es decir, deben ser empacados con su arista w y h paralelas a la arista W y H del de la superficie del contenedor, o pueden ser rotados 90° .
- *Carga secuencial:* Se puede imponer que el patrón de carga de cada vehículo permita que la demanda de ítems de cada cliente pueda ser descargada a través de una secuencia de un movimiento directo por ítem paralelo al eje H de la carga de área. Esta versión es denotada como carga secuencial, donde los ítems pueden ser retirados al visitar cada cliente sin requerir la reubicación del resto de la carga, mientras que la versión en la que no está impuesta la restricción de carga secuencial donde el patrón de carga puede requerir modificaciones cada vez que se visita un cliente, es comúnmente conocida como irrestricta.

Nótese que la restricción de carga secuencial impone que el corredor que va desde la arista w de cualquier ítem demandado por un cliente, hasta la parte trasera del vehículo, no puede contener ninguna porción o ítem demandado por los clientes que son visitados posteriormente en la misma ruta.

Carga tridimensional en los vehículos 3L-CVRP

Muchas actividades en el transporte de mercancías involucran dos tópicos básicos de optimización que han sido estudiados intensamente en las últimas décadas. El primero,

denominado ruteo óptimo, consiste en encontrar las rutas que conectan la secuencia de clientes a ser visitados para entregar los bienes a mínimo costo, y el segundo, determinar la mejor manera de acomodar dichos bienes en los vehículos empleados para el transporte. La gran mayoría de los problemas que surgen en esas dos áreas pertenecen a la clase de los problemas NP-duros, y en la práctica su solución es realmente desafiante.

El problema de enrutamiento de vehículos surge cuando es necesario repartir un conjunto de bienes a una serie de clientes que se encuentran geográficamente dispersos, el objetivo consiste en minimizar los costos de la distribución o maximizar las ganancias asociadas al transporte de las demandas que se deben atender. Por otra parte, el problema de carga de contenedores o empaquetamiento es un problema de optimización tridimensional en donde se debe acomodar un conjunto de elementos dentro de una caja rectangular de gran tamaño, de forma tal que se optimice alguna o varias funciones objetivo, satisfaciendo ciertas restricciones de empaquetamiento.

Aplicaciones reales de este problema ocurren en muchos contextos industriales, tales como: el transporte de aglomerados de madera para muebles, empresas de envío de paquetes, transporte de vehículos y transporte de mercancía en *pallets*, entre otros. El enrutamiento y el empaquetamiento óptimo no solo es un requisito económico, sino que también se refiere a la realización de una distribución más eficiente que disminuya el tamaño de las rutas, reduciendo el impacto ambiental asociado a los vehículos que utilizan los planes de entrega.

La variante de enrutamiento CVRP busca un conjunto de circuitos o llamados rutas iniciando y terminando en el depósito central. El CVRP se define como K vehículos idénticos de capacidad D que deben atender n clientes diferentes, en donde cada vehículo es asignado por lo menos a una ruta, cada cliente es visitado por un vehículo, la suma de los volúmenes de las cajas demandadas de cada ruta no es mayor a la capacidad D del vehículo y el costo general de todas las rutas debe ser la mínima para satisfacer las condiciones anteriores. Por otro lado, el problema de empaquetamiento que se debe resolver para cada ruta propuesta, consiste en cargar dentro de un contenedor pequeños paralelepípedos o cajas, de tamaños diferentes y cantidades limitadas, de forma tal que se maximiza el espacio ocupado por la

carga. Este problema en la literatura es clasificado como el problema de emplazamiento tridimensional en un único gran objeto rectangular (3D-SLOPP o CLP, por sus acrónimos en inglés *Three-dimensional Rectangular Single Large Object Placement Problem* y *Container Loading Problem*, respectivamente).

La integración del problema de enrutamiento de vehículos capacitado CVRP y la carga de piezas tridimensionales en un único gran objeto rectangular 3D-SLOPP da origen al problema conocido como 3L-CVRP, es decir, problema de enrutamiento de vehículos capacitado con restricciones de carga de contenedores.

Además de esto, dentro de la validación del montaje de la carga en el vehículo, pueden ser también consideradas las restricciones de: orientación y fragilidad de las cajas, estabilidad de la carga, y secuenciamiento de la carga. Dependiendo de las diferentes restricciones de empaquetamiento requeridas por el contexto, se obtienen diferentes variantes del problema integrado. En este trabajo es desarrollado un algoritmo de solución para la variante donde la capacidad D del vehículo es igual a su volumen, ya que en la práctica los vehículos están limitados únicamente por sus dimensiones. La metodología propuesta obtiene unos resultados aceptables para la versión 3L-CVRP.

El 3L-CVRP considera que el peso total d_i de demanda del cliente i ($i = 1, 2, \dots, n$) es producido por m_i ítems tridimensionales. Cada ítem I_{il} ($l = 1, 2, \dots, m$), tiene ancho w_{il} , alto h_{il} y largo l_{il} . La superficie de carga de cada vehículo tiene ancho W , alto H y largo L . Sea $S(k) \subseteq \{1, 2, \dots, n\}$ el conjunto de clientes visitados por el vehículo k . Además de la restricción estándar de capacidad, una restricción de carga impone que existe un empaquetamiento factible sin ningún tipo de traslape de todos los ítems tridimensionales solicitados por cada uno de los clientes de $S(k)$ dentro del espacio de carga de dimensiones $W \times H \times L$. Así como para el caso bidimensional, un número adicional de restricciones prácticas o reales, aparecen en la literatura:

- *Orientación:* los ítems pueden tener orientación fija, o (con mucha más frecuencia) pueden ser rotados 90° en el plano horizontal (mientras que la rotación de orientación vertical usualmente no se permite).

- *Fragilidad*: cada ítem I_{il} puede presentar una bandera de fragilidad f_{il} , tomando valor 1 si I_{il} es frágil, y valor 0 de lo contrario. En este caso, una restricción adicional impone que los ítems que no son frágiles no pueden ser ubicados sobre ítems frágiles.
- *Área de soporte*: Cada ítem I_{il} puede ser empacado sobre otros ítems. Sea \bar{A} el área del fondo del ítem o producto I_{il} . El empaquetamiento es factible sí y solo sí, $\bar{A} \geq a \cdot w_{il} \cdot l_{il}$, donde a es un umbral dado ($0 \leq a \leq 1$) y representa la porción mínima del área de la caja que tiene contacto con la caja de posible soporte del ítem I_{il} .
- *Carga secuencial*: cuando un cliente i es visitado, debe existir una secuencia de movimientos rectos (uno por cada ítem I_{il}) en la dirección de la parte trasera del vehículo, que permite la descarga del ítem sin mover ningún otro ítem. En otras palabras, ningún ítem demandado por un cliente que sea visitado después puede ser ubicado sobre I_{il} o entre I_{il} y la parte trasera del vehículo.

Este conjunto de restricciones de empaquetamiento son modeladas y enlazadas con el modelo de transporte en (Junqueira et al., 2013). En este trabajo, estas restricciones no son adicionadas al modelo debido a que serán garantizadas a través de un algoritmo heurístico.

2.3.2. Definición del problema de enrutamiento de vehículos considerando impacto ambiental

Se han planteado diferentes maneras de involucrar el impacto ambiental en los modelos matemáticos de enrutamiento de vehículos, de manera general se presentan a continuación:

- *Modelos con factor de contaminación*: La carga (demanda) como factor del costo de los trayectos que debe viajar el vehículo con los bienes.
- *Modelos de velocidad promedio*: Velocidad promedio de viaje del vehículo entre clientes o puntos de atención como parte del conjunto de variables de decisión para medir la contaminación, no solo basándose en la carga que lleva el vehículo, si no también la velocidad a la que la transporta.

- *Modelos instantáneos:* Velocidad instantánea y aceleración entre los puntos de atención, considerando elementos externos al viaje del vehículo, tales como, congestión en las vías asociada a horas pico, accidentes y obstáculos en el trayecto para entregar la demanda transportada.

Cada uno de estos enfoques es incremental, y son enumerados desde el más sencillo hasta el que presenta más elementos para modelar la participación del transporte de bienes en el deterioro del medio ambiente. Por lo tanto, todos se basan en la carga o demanda acumulada que transportan de los clientes mientras se desplazan entre los puntos geográficamente dispersos que deben visitar, adicionando elementos como la velocidad a la que viajan o los cambios de la misma dependiendo del entorno (congestión vehicular, accidentes). Estos elementos adicionales requieren de una toma exhaustiva de información durante el trayecto, necesitando equipos muy costosos y limitando el análisis a trayectos más cortos y áreas más reducidas por el volumen de información que podría requerirse, esto sería más cercano a estudios de simulación que a optimización y análisis de este subgrupo de problemas de enrutamiento de vehículos.

Capítulo 3

Algoritmo mateheurístico para el problema de enrutamiento de vehículos con flota homogénea y restricciones de empaquetamiento tridimensionales 3L-CVRP

3.1. Introducción

En este capítulo se presenta un algoritmo híbrido para resolver el problema de ruteo de vehículos con restricciones de capacidad y restricciones prácticas de empaquetamiento tridimensional 3L-CVRP. La metodología de solución propuesta en este trabajo consiste de dos fases; la primera utiliza un procedimiento de optimización basado en cortes para el CVRP, la segunda valida las soluciones de la fase anterior a través de un algoritmo GRASP, que evalúa las restricciones de empaquetamiento de cada una de las rutas. Para el algoritmo híbrido se utiliza la relajación del modelo clásico de dos subíndices para el problema de

ruteo de vehiculos. En particular diferentes tipos de cortes son adicionados: eliminacion de subtours, cortes debido a las restricciones de capacidad y cortes para restricciones de empaquetamiento. El algoritmo propuesto ha sido comparado con los algoritmos mas eficaces para el 3L-CVRP en el conjunto clasico de instancias presentadas en la literatura. Los resultados computacionales muestran que el metodo propuesto es capaz de obtener buenos resultados perfeccionando algunas de las mejores soluciones conocidas propuestas en la literatura.

3.2. Modelamiento del 3L-CVRP

3.2.1. Problema de Enrutamiento de Vehículos Capacitado CVRP

En el CVRP se busca obtener un número específico de ciclos (rutas) que satisfagan la demanda de un conjunto de vértices (clientes) iniciando y terminando en un depósito central ubicado en el vértice 0. La formulación completa del CVRP propuesta en (Toth and Vigo, 2002), la cual es ampliamente conocida como formulación de dos índices es presentada de la siguiente manera. Sea $G = (V, A)$ un grafo completo, donde $V = 0, 1, 2, \dots, n$ es el conjunto de vértices y A el conjunto de arcos. Los vértices $V = 0, 1, 2, \dots, n$ corresponden a los clientes, donde el vértice 0 corresponde al depósito. Un costo no-negativo de viaje c_{ij} es asociado con cada arco $(i, j) \in A$. El costo de viaje entre (i, i) no es permitido. Por lo tanto, el costo $c_{ii} = +\infty \forall i \in V$. Específicamente, este trabajo considera la versión simétrica del CVRP. Por tal razón, $c_{ij} = c_{ji} \forall (i, j) \in A$, y el conjunto de arcos puede ser reemplazado por un conjunto de aristas no dirigidas E . Cada vértice $i \in V$ es asociado con una demanda no-negativa conocida d_i que debe ser entregada. Nótese que el depósito tiene una demanda ficticia $d_0 = 0$. Dada una arista $e \in E$, sea $\alpha(e)$ y $\beta(e)$ sus vértices terminales. Dado un conjunto de vértices $S \subseteq V$, sea $\delta(S)$ y $E(S)$ el conjunto de aristas $e \in E$ que tienen uno o ambos puntos terminales en S respectivamente. Adicionalmente, se denota $d(S) = \sum_{i \in S} d_i$ como la demanda total del conjunto S .

Un conjunto K de vehículos idénticos, cada uno con capacidad Q , están disponibles en el

depósito. Para asegurar la factibilidad se asume que $d_i \leq Q$ para cada $i = 1, \dots, n$. Cada vehículo realiza exactamente una ruta. Para un conjunto $S \subseteq V \setminus 0$, denotado por $r(S)$ el número mínimo de vehículos necesario para servir a todos los clientes en S . Comúnmente, $r(S)$ es reemplazado por el límite inferior obtenido por el Problema de Empaquetamiento de Contenedores (BPP por sus siglas en inglés *Bin Packing Problem*) trivial asociado al problema: $\left\lceil \frac{d(S)}{Q} \right\rceil$. El BPP permite determinar el número mínimo de contenedores (vehículos), cada uno con capacidad Q , requeridos para cargar los n ítems, cada uno con un peso no-negativo $d_i (i = 1, 2, \dots, n)$ siendo NP-duro estrictamente.

El CVRP consiste en encontrar un conjunto K de rutas con costo mínimo, definido como la suma de los costos de los arcos que pertenecen a las rutas que deben realizarse. El CVRP está sujeto a las siguientes restricciones:

1. Cada ruta termina e inicia en el vértice correspondiente al depósito.
2. Cada vértice cliente es visitado exactamente una vez
3. La suma de las demandas asociadas a los vértices visitados en una ruta no puede exceder la capacidad Q de los vehículos.

El modelo utilizado en este capítulo, es una formulación de flujo de dos índices que utiliza variables binarias x para indicar si un vehículo viaja a través de un arco (i, j) en la solución óptima (Paschos, 2013). En otras palabras, la variable x_{ij} toma el valor de 1 si el arco $(i, j) \in E$ pertenece a la solución óptima y toma el valor de 0 de lo contrario. La función objetivo consiste en minimizar el costo de los arcos a través de los que se viaja, como se presenta en la Ecuación 3.1. Las Ecuaciones (3.2 - 3.6) controlan las visitas a los clientes y la eliminación de subtours.

$$\text{Min} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}; \quad (3.1)$$

s.a.

$$\sum_{i \in V} x_{ij} = 1, \forall j \in V \setminus \{0\} \quad (3.2)$$

$$\sum_{j \in V} x_{ij} = 1, \forall i \in V \setminus \{0\} \quad (3.3)$$

$$\sum_{j \in V} x_{0j} = K \quad (3.4)$$

$$\sum_{i \in V} x_{i0} = K \quad (3.5)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S), \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (3.6)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V \quad (3.7)$$

Las restricciones 3.2 y 3.3 corresponden a las restricciones de flujo del conjunto V (clientes); estas restricciones garantizan que cada cliente deba ser visitado por un vehículo una única vez. Las restricciones 3.4 y 3.5 aseguran que el mismo número de rutas lleguen y partan del depósito 0. El conjunto 3.6 considera las restricciones de eliminación de subtours. De acuerdo a (Toth and Vigo, 2002), este conjunto de restricciones puede ser representado de la siguiente manera en la Ecuación 3.8:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S) \quad (3.8)$$

Sin embargo, el conjunto de restricciones encargadas de la eliminación de subtours requiere consideraciones especiales debido a su complejidad combinatorial cuando el número de clientes se incrementa. Por lo tanto, se ha considerado el conjunto de restricciones 3.8 para el algoritmo propuesto. En particular, se añaden los cortes requeridos para eliminar los subtours, hasta que una solución factible es encontrada durante el procedimiento *Branch-and-Bound*.

Nótese que el modelo considerado es capaz de representar solamente el CVRP. Para

obtener una representación global del problema 3L-CVRP, se ha considerado el conjunto de restricciones 3.9, las cuales son adicionadas de manera iterativa al mismo tiempo que las restricciones de eliminación de subtours. Este conjunto de restricciones garantiza la factibilidad de los requerimientos de empaquetamiento (como las restricciones de carga secuencial o *multidrop*, entre otras). Sea $U \subseteq V \setminus \{0\}$ un subconjunto de clientes con una demanda acumulada, la cual no puede ser empacada en los contenedores de los vehículos considerando restricciones de empaquetamiento *multidrop*. Todos los posibles subconjuntos de clientes que no pueden ser empacados serán entonces controlados por la restricción 3.9:

$$\sum_{i \in U} \sum_{j \in U} x_{ij} \leq |U| - 2 \quad (3.9)$$

Adicionando este conjunto de restricciones, el modelo matemático puede representar apropiadamente el problema 3L-CVRP. El nivel de dificultad de solución del modelo presentado en (3.1 - 3.9), se incrementa cuando el número de nodos es mayor, debido a la explosión combinatorial que surge de los posibles subconjuntos de clientes controlado por 3.8. La alta complejidad del CVRP ha llevado al desarrollo de varios algoritmos basados tanto en métodos exactos como aproximados. El CVRP ha sido investigado desde la década del 50. Artículos de revisión del CVRP pueden encontrarse en (Toth and Vigo, 2002), (Golden et al., 2008) y (Baldacci et al., 2010). En (Schmid et al., 2013), un problema de enrutamiento de vehículos que surge en la gestión de cadena de suministros es propuesto, incluyendo el 3L-CVRP.

3.2.2. Problema de Carga y Empaquetamiento

En situaciones realistas de carga y empaquetamiento, la demanda de los clientes no puede ser caracterizada simplemente como una cantidad (como el caso del CVRP), si no que también es determinada por su forma y su ubicación espacial. En este caso, es necesario asegurar que debe ser entregado, debe ser ubicado dentro del contenedor del vehículo. Estas restricciones conciernen a los problemas de empaquetamiento multidimensional rectangular, los cuales se originan como una extensión del problema de carga de contenedores unidimensional BPP. El

BPP puede ser descrito como el problema de ubicar un conjunto de segmentos sin que estos se traslapen. Una introducción general a los trabajos correspondientes al empaquetamiento rectangular puede encontrarse en los trabajos: (Lodi et al., 2002), (Paschos, 2013), (Wascher et al., 2007) y (E.G. Coffman et al., 2013).

3.3. Definición del Problema

El 3L-CVRP considera que ítems tridimensionales generan el peso total de la demanda de un cliente i ($i = 1, 2, \dots, n$). Cada ítem I_{il} ($l = 1, 2, \dots, m$) tiene un ancho w_{il} , un alto h_{il} y un largo l_{il} . El área de carga de cada vehículo tiene un ancho W , un alto H y una longitud L . Sea $S(k) \subseteq \{1, 2, \dots, n\}$ el conjunto de clientes visitados por el vehículo k . El 3L-CVRP impone las restricciones de empaquetamiento mencionadas en la Sección 2.3.1.

Este trabajo considera todas las restricciones de empaquetamiento, caracterizándolas como se presentan en 2.3.1. Es importante notar que esta caracterización presenta suposiciones que limitan su funcionalidad y aplicabilidad en contextos reales. Por ejemplo, la restricción de fragilidad podría ser formulada como una expresión binaria dependiendo de la fuerza de soporte de carga y la orientación de las cajas (Alonso et al., 2014).

Los trabajos publicados previos a este estudio para el 3L-CVRP han intentado eliminar algunas restricciones de empaquetamiento, con el fin de aislar las más críticas. En (Martínez et al., 2015), los autores indican que la secuencia de carga normalmente es la restricción dominante. En este trabajo, otros aspectos del problema son estudiados; particularmente, el hecho de que la capacidad Q del vehículo es normalmente especificada como parámetro sin que presente ninguna relación con el tipo de carga. Esto implicaría que el valor de $d(i)$ (demanda de cada cliente) asume que todas las cajas del cliente i , e incluso para el conjunto total de clientes, presentan la misma densidad de material, es decir, todos los clientes, teniendo diferentes tipos de cajas, supuestamente tendrían la misma densidad de material, situación que es muy poco probable que se presente en la práctica.

Hay dos tipos de suposiciones acerca de las restricciones de capacidad del vehículo:

- Transporte de todo tipo de densidades de material para las cajas de cada uno de los clientes.
- Transporte de un tipo de densidad de material de las cajas (el cual se asume como 1, siendo el peso de cada caja igual a su volumen).

Los trabajos publicados previamente solamente consideran la primera suposición. Se han considerado ambas suposiciones (mirar resultados de la columna correspondiente a la Mateheurística y la columna de la Mateheurística 3L-VRP presentes en la Tabla 3.1). De hecho, una contribución de este trabajo es examinar el efecto de tener una densidad igual a 1. Consecuentemente, para la segunda suposición, el nuevo valor de $d(S)$ sería $d(S) = \sum_{i \in S} w_{il} \cdot l_{il} \cdot h_{il}$ y la nueva capacidad del vehículo correspondería a $Q = W \times L \times H$.

Todos los conjuntos de restricciones de empaquetamiento son integradas y enlazadas con el modelo de transporte presentado en (Junqueira et al., 2013), pero debido a la alta complejidad del modelo, la metodología propuesta es inadecuada para resolver problemas de mediano y gran tamaño. En este trabajo se utiliza un algoritmo GRASP dentro del modelo matemático, garantizando dichas restricciones, y permitiendo solucionar problemas realistas que se presentan en empresas dedicadas a estas actividades. Muchos algoritmos aproximados han sido propuestos para resolver el 3L-CVRP.

Nótese que todos los enfoques propuestos para el 3L-CVRP están basados en esquemas heurísticos, exceptuando (Bortfeldt, 2012). En este estudio se propone un algoritmo mateheurístico, el cual difiere de (Bortfeldt, 2012), porque el problema de enrutamiento es resuelto por un método exacto y el problema de empaquetamiento por un algoritmo aproximado. El algoritmo propuesto es ampliado en las siguientes secciones.

3.4. Enfoque Mateheurístico

La estrategia de solución general propuesta, resuelve ambos problemas de manera separada: el Problema Capacitado de Enrutamiento de Vehículos (CVRP), y el Problema Tridimensional

de Carga de Contenedores Secuencial (3D-SLOPP, por sus siglas en inglés: *Sequential Loading Packing Problem*). Para cada solución del CVRP, una validación de las restricciones de empaquetamiento de la carga del contenedor para cada ruta es realizada. La principal fortaleza del abordaje propuesto, es que el esfuerzo computacional está principalmente enfocado en la solución exacta del CVRP, mientras que el problema de carga es resuelto con un GRASP de alto rendimiento. El GRASP es calibrado de acuerdo a las características de los ítems que se entregarán, es decir, la demanda acumulada de los clientes cubiertos en cada ruta.

El modelo matemático de dos índices (3.1 - 3.9) es relajado eliminando las restricciones de empacabilidad y control de subtours. El enfoque propuesto gradualmente inserta esas restricciones durante el esquema *Branch-and-Cut*, con el fin de obtener soluciones factibles. De hecho, el algoritmo propuesto, empieza con una solución inicial generada por el conocido algoritmo de ahorros (Clarke and Wright, 1964) y validado por el algoritmo GRASP. El valor de la función objetivo de la solución inicial es utilizado como límite superior (*upper bound*) de la metodología propuesta.

El algoritmo permite soluciones infactibles para el 3L-CVRP debido a que la primera solución factible encontrada durante la búsqueda corresponde al óptimo del problema. Cuando una solución factible es encontrada para el problema de enrutamiento, esta provee un límite inferior (*lower bound*) para el problema original. Por esta razón, se puede utilizar como inicialización del 3L-CVRP durante la búsqueda. El límite superior está dado por la demanda cargada correspondiente para las K rutas que se deben realizar para la distribución de la carga (desde esta perspectiva el problema es visto como un problema de *bin-packing*). Sin embargo, el esfuerzo del algoritmo propuesto está totalmente orientado a encontrar una solución factible para el problema de enrutamiento de vehículos, guiado por la minimización de los costos totales de viaje. Aunque el modelo de dos índices (3.1 - 3.9) requiere un esfuerzo computacional notable, permite articular adecuadamente las restricciones correspondientes al manejo de la carga de las cajas en el contenedor.

El 3L-CVRP ha sido direccionado aplicando un método exacto (*Branch-and-Cut*) para el

problema de enrutamiento de vehículos, resuelto a través de ILOG *Concert Technology* y CPLEX bajo C++. El problema de empaquetamiento es resuelto con un GRASP. Inicialmente, la versión relajada del modelo (3.1 - 3.9) es resuelto sin las restricciones de capacidad y eliminación de subtours. Luego, esas restricciones son adicionadas de manera iterativa durante el proceso. El algoritmo termina cuando una solución es encontrada para el modelo, involucrando todas la restricciones. El algoritmo mateheurístico es descrito a continuación.

3.4.1. Método exacto para el Problema de Enrutamiento de Vehículos

En el algoritmo propuesto, se ha considerado el modelo de dos índices (3.1 - 3.9) para resolver el subproblema de CVRP. Esta formulación está basada en subconjuntos de S clientes para controlar la aparición de subtours y las restricciones de capacidad de los vehículos. Las restricciones (3.6) y (3.8) son eliminadas obteniendo un modelo relajado que se denotará M_{CVRP} .

La metodología propuesta inicia resolviendo el M_{CVRP} y guardando su solución óptima en S . Luego, la solución óptima actual para el M_{CVRP} es revisada para encontrar las violaciones sobre las restricciones de capacidad L_u o la presencia de subtours L_S . Si la solución es infactible, los cortes correspondientes son añadidos por la función $AddCutsToTheModel(M_{CVRP}, L_S, L_e, R_i)$ para evitar estas infactibilidades en las siguientes iteraciones. Si la solución es factible, entonces el procedimiento debe revisar el empaquetamiento correspondiente a la demanda acumulada en cada ruta a través del GRASP, el cual determina si es posible o no empaquetar las cajas en los contenedores de los vehículos con un patrón de carga que no requiera reorganización de las mismas en cada visita a los clientes. Además, si la ruta inversa (posición inversa de los clientes) es imposible de empaquetar, las rutas consideradas deben ser prohibidas (L_u). En el Algoritmo 1 se detalla el proceso.

Con el fin de evitar la aparición de rutas violando los conjuntos de restricciones, los cortes $\sum_{i \in U} \sum_{j \in U} x_{ij} \leq |U| - 2$ (3.9) son añadidos al modelo M_{CVRP} . Las aristas de las rutas con clientes perteneciendo a U (subconjunto de clientes que juntos configuran una ruta

Algorithm 1 Matheuristic Branch&Cut-GRASP

Data: CVRP Mathematical Model M_{CVRP} , Initial solution with a modified Clarke&Wright Savings

(packing feasible) R_0 , 3L-CVRP Solution R_s , Subtours List L_s , List of routes with unfeasible packing L_u , Time of execution for the incremental model in seconds T_e

Result: Optimal and feasible solution R_s for the 3L-CVRP

```
1  $R_0 \leftarrow$  ModifiedSavings;  
   Initialize mathematical model  $M_{CVRP}$  without the constraints with combinatorial explosion;  
   Start timer  $T_e$ ;  
   while ( $L_s \neq \emptyset$  or  $L_u \neq \emptyset$ ) and  $T_e < 3600$  do  
2    $L_s \leftarrow \emptyset, L_u \leftarrow \emptyset$ ;  
      $R_s \leftarrow$  Solve( $M_{CVRP}$ , Starting point  $R_0$ );  
      $L_s \leftarrow$  GetSubtours( $R_s$ );  
     if  $L_s = \emptyset$  then  
3       for Each Route  $r$  of  $R_s$  do  
4          $L_u \leftarrow$  GRASP( $r$ );  
5       end  
6     end  
7     AddCutsToTheModel( $M_{CVRP}, L_s, L_u$ );  
8 end  
9 return  $R_s$ ;
```

infactible en términos de empaquetamiento) en una determinada secuencia serán restringidas. Sin embargo, muchas secuencias de clientes serán evaluadas para una ruta dada, intentando encontrar soluciones factibles considerando las restricciones de empaquetamiento. La Figura 3.1 ilustra los cortes de empaquetamiento realizados.

Una solución relajada para el M_{CVRP} relajado, considera subtours y también rutas que son infactibles con respecto a las restricciones de empaquetamiento (ver Ruta a en la Figura 3.1). Los clientes que pertenecen a la Ruta a no pueden ser empacados porque se viola la restricción de carga secuencial, es decir, la ruta $0 - 1 - 5 - 8 - 11 - 7 - 0$ debe ser eliminada. Sin embargo, una gran cantidad de permutaciones de la Ruta a ($0 - 7 - 11 - 8 - 5 - 1 - 0$

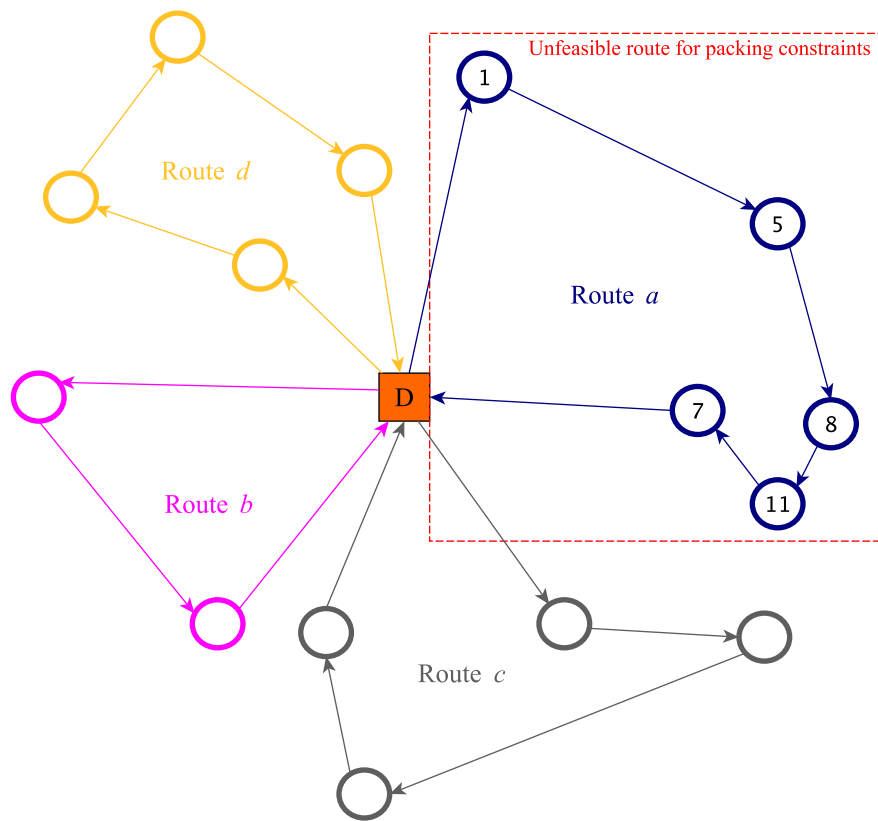


Figura 3.1: Solución obtenida con el modelo relajado

por ejemplo) requieren ser examinadas en iteraciones posteriores, porque podrían brindar soluciones factibles en términos de empaquetamiento. Por lo tanto, los clientes perteneciendo a la Ruta a infactible, configuran el conjunto $U = \{1, 5, 8, 11, 7\}$, $U \subseteq V \setminus \{0\}$ y $|U| = 5$. Luego, a partir de la Ecuación 3.9, el corte expresado en la Ecuación 3.10 es aplicado.

$$x_{1\ 5} + x_{5\ 8} + x_{8\ 11} + x_{11\ 7} \leq 5 - 2$$

$$x_{1\ 5} + x_{5\ 8} + x_{8\ 11} + x_{11\ 7} \leq 3 \quad (3.10)$$

Consecuentemente, cualquier conjunto de las aristas involucradas en la Ecuación 3.10 es permitido, pero la secuencia completa de la ruta en el orden inicial, es restringida con la introducción del corte 3.10 en el modelo M_{CVRP} para las siguientes iteraciones. El proceso continúa iterando en soluciones factibles para el problema de enrutamiento de vehículos, pero infactible para el problema de carga secuencial de contenedores tridimensional. La solución óptima es encontrada cuando la carga completa de las rutas de la solución óptima para el M_{CVRP} , sea también factible para el 3D-SLOPP.

3.4.2. GRASP para validación de empaquetamiento

En este trabajo, se utiliza el algoritmo GRASP, por sus siglas en inglés *Greedy Randomized Adaptive Search Procedure*, presentado en (Feo and Resende, 1989). El enfoque trabajado en este estudio está basado en espacios maximales, en este caso tridimensionales como se presenta en la Figura 3.2.

Más adelante, en la Sección 4.2.2 se detalla el algoritmo aproximado de empaquetamiento tridimensional utilizado en este trabajo.

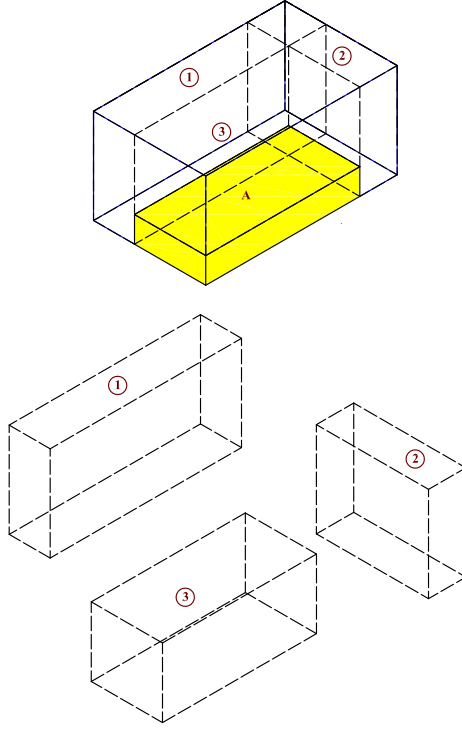


Figura 3.2: Ubicación de una caja en un subespacio maximal, generando nuevos subespacios maximales

3.5. Resultados Computacionales

Se han considerado el conjunto de instancias clásicas (27) instancias para validar el desempeño de la metodología propuesta. La mateheurística presentada ha sido comparada con (Gendreau et al., 2006), (Tarantilis et al., 2009), (Fuellerer et al., 2010), (Ruan et al., 2013), (Bortfeldt, 2012) y (Tao and Wang, 2015). Varias mejores soluciones conocidas han sido mejoradas. Los tiempos computacionales de la metodología propuesta son altos comparados con los abordajes publicados. El conjunto de instancias de prueba para el 3L-CVRP han sido adquiridas de la librería publicada en http://or.dei.unibo.it/sites/or.dei.unibo.it/files/instances/3l_cvrp.zip. La Tabla 3.1 muestra el costo total de las rutas realizadas para entregar todas las cajas a los clientes (función objetivo del 3L-CVRP). La Figura 3.3 muestra las rutas obtenidas para el algoritmo propuesto. Las Figuras 3.4-3.7 muestran los patrones de empaquetamiento correspondientes a las rutas solución de la Instancia 1. Como se observa, los patrones de empaquetamiento satisfacen las restricciones de estabilidad estática

de la carga, fragilidad de las cajas y descarga sin reacomodar la carga (política *multi-drop* o LIFO).

Tabla 3.1: Comparación con los mejores resultados de la literatura

Instancia	Nodos	Ítems	Vehículos	BKS	Gendreau et al. (2006)		Tarantilis et al. (2009)		Fuellerer et al. (2010)			Bortfeldt (2012)			Ruan et al. (2013)		Tao and Wang (2015)			Mateheurística		Mateheurística (3L-VRP)	
					Solución	Tiempo	Solución	Tiempo	Mejor	Promedio	Tiempo	Mejor	Promedio	Tiempo	Solución	Tiempo	Mejor	Promedio	Tiempo	Solución	Tiempo	Solución	Tiempo
1	15	32	4	302,02	316,32	1800	321,47	13,2	304,13	305,35	12	302,02	302,02	72,3	303,21	98,85	302,02	302,02	53,5	300,69	492,9	296,03	107
2	15	26	5	334,96	350,58	1800	334,96	11,5	334,96	334,96	0,6	334,96	334,96	0,9	334,96	4,55	334,96	334,96	6,3	359,78	941,1	292,17	7
3	20	37	4	381,37	447,73	1800	430,95	540,6	399,68	409,79	121,8	392,63	401,44	182	398,05	93,86	381,37	381,37	116,2	374,81	228,7	347,69	154
4	20	36	6	437,19	448,48	1800	458,04	323,5	440,68	440,68	5,4	437,19	437,54	16,1	440,68	46,75	437,19	437,19	14	430,88	492,9	360,91	8
5	21	45	6	436,48	464,24	1800	465,79	99,6	450,93	453,19	30,9	443,61	451,03	183	452,56	63,98	436,48	436,79	149,3	495,32	495,3	495,32	3600
6	21	40	6	498,16	504,46	1800	507,96	1212	498,32	501,47	18,4	498,16	498,38	23,6	498,56	196,9	498,32	498,32	31,6	498,31	425,7	416,15	17
7	22	46	6	767,46	831,66	1800	796,61	364,8	792,13	797,47	67,4	769,68	772,49	133	790,23	317,02	767,46	768,94	84,9	845,51	3600	740,15	335
8	22	43	6	804,75	871,77	1800	880,93	230	820,67	820,67	78,6	810,89	821,35	139	820,67	98,9	804,75	805,77	120,4	942,57	3600	752,37	496
9	25	50	8	630,13	666,1	1800	642,22	982,2	635,5	635,5	16,3	630,13	645,81	24,3	635,5	353,07	630,13	631,68	13,7	639,25	3600	547,28	78
10	29	62	8	820,35	911,16	3600	884,74	1308	840,75	841,12	246,7	820,35	827,29	175	836,21	410,9	822,86	828,99	258,6	906,2	3600	906,2	3600
11	29	58	8	772,85	819,36	3600	873,43	522,5	818,87	821,04	199,8	803,61	815,62	136	825,75	197,76	772,85	780,61	278,9	840,44	3600	840,44	3600
12	30	63	9	614,59	651,58	3600	624,24	294,6	626,37	629,07	48,2	614,59	630,46	14	626,59	89,47	614,59	614,6	145,8	641,3	3600	641,3	3600
13	32	61	8	2608,7	2928,34	3600	2799,74	2193	2739,8	2739,8	308,8	2646	2694,81	268	2739,8	319,78	2608,7	2636,85	369,4	2894,33	3600	2894,33	3600
14	32	72	9	1368,4	1559,64	3600	1504,44	4581	1466,8	1472,26	642,8	1368,4	1413,59	312	1469,38	268,39	1391,6	1398,77	588,1	1673,19	3600	1673,19	3600
15	32	68	9	1341,1	1452,34	3600	1415,42	2528	1367,6	1405,48	656,8	1341,1	1355,5	312	1369,69	356,55	1345,7	1352,76	615,9	1498,51	3600	1498,51	3600
16	35	63	11	698,61	707,85	3600	698,61	4257	698,92	698,92	14,8	698,61	705,05	3,4	703,15	431,74	698,92	698,92	4	731,46	3600	731,46	3600
17	40	79	14	866,4	920,87	3600	872,79	2096	868,59	870,33	14,9	866,4	917,96	2,5	872,05	374,84	866,4	866,4	9,5	881,53	3600	881,53	3600
18	44	94	11	1207,7	1400,52	3600	1296,59	2275	1255,6	1261,07	2209,8	1207,7	1228,98	310	1250,86	325,74	1220,3	1228,47	1634	1415,46	3600	1415,46	3600
19	50	99	12	741,74	871,29	7200	818,68	2509	777,18	781,29	623,6	741,74	753,87	417	780,37	1374,8	760,97	763,09	718,4	843,4	3600	843,4	3600
20	71	147	18	587,95	732,12	7200	641,57	1941	604,28	611,26	3901	587,95	596,42	427	605,59	1337	589,66	590,99	2942	659,84	3600	659,84	3600
21	75	155	17	1086,2	1275,2	7200	1159,72	2823	1110,1	1124,55	5180,6	1090,2	1107	443	1119,45	1247,9	1086,2	1096,53	2301	1181,55	3600	1181,55	3600
22	75	146	18	1147,8	1277,94	7200	1245,35	2686	1194,2	1197,43	2290,3	1147,8	1171,49	424	1167,28	1294,6	1150,6	1155,81	1242	1242,84	3600	1242,84	3600
23	75	150	17	1127,9	1258,16	7200	1231,92	4659	1158,5	1171,77	3727,6	1130,5	1135,46	426	1171,77	1105,7	1127,9	1130,08	1925	1159,44	3600	1159,44	3600
24	75	143	16	1114,1	1307,09	7200	1201,96	4854	1136,8	1148,7	1791,5	1116,1	1128,82	411	1136,27	2001,1	1114,1	1122,8	2527	1222,98	3600	1222,98	3600
25	100	193	22	1407,4	1570,72	7200	1457,46	5726	1429,6	1436,32	8817,1	1407,4	1428,8	453	1426,34	1458,8	1415,8	1417,09	4536	1507,89	3600	1507,89	3600
26	100	199	26	1585,5	1847,95	7200	1711,93	6283	1611,8	1616,99	6904,3	1600,4	1625,31	431	1585,46	3354,7	1602,1	1605,11	3018	1832,86	3600	1832,86	3600
27	100	198	23	1529,9	1747,52	7200	1646,44	9916	1560,7	1573,5	10484	1529,9	1550,85	435	1562,18	3140,2	1535,8	1538,1	6026	1647,88	3600	1647,88	3600
Promedio (Distancia Total)				934,06	1042,26		997,18		966,67			953,79			960,1		941,59			1024,75		1001,08	
GAP Relativo % (3L-VRP)					-1,68		2,76		6,01			7,44			6,73		8,83			-		-	
GAP Relativo % (3L-CVRP)					-3,95		0,39		3,56			4,96			4,27		6,32			-		-	
Tiempo Promedio (s)					4200		2416		1793			229			754		1101			2913		2577	
CPU					Pent. IV 3.0 GHz		Pent. IV 2.8 GHz		Pentium IV 3.2 GHz			Core2Duo E8500 3.16GHz			Pentium IV 2.3 GHz		Pentium IV 2.67 GHz			i7-2600S 2.8GHz		i7-2600S 2.8GHz	
Single Thread Rating					653		635		699			1321			564		608			1797		1797	

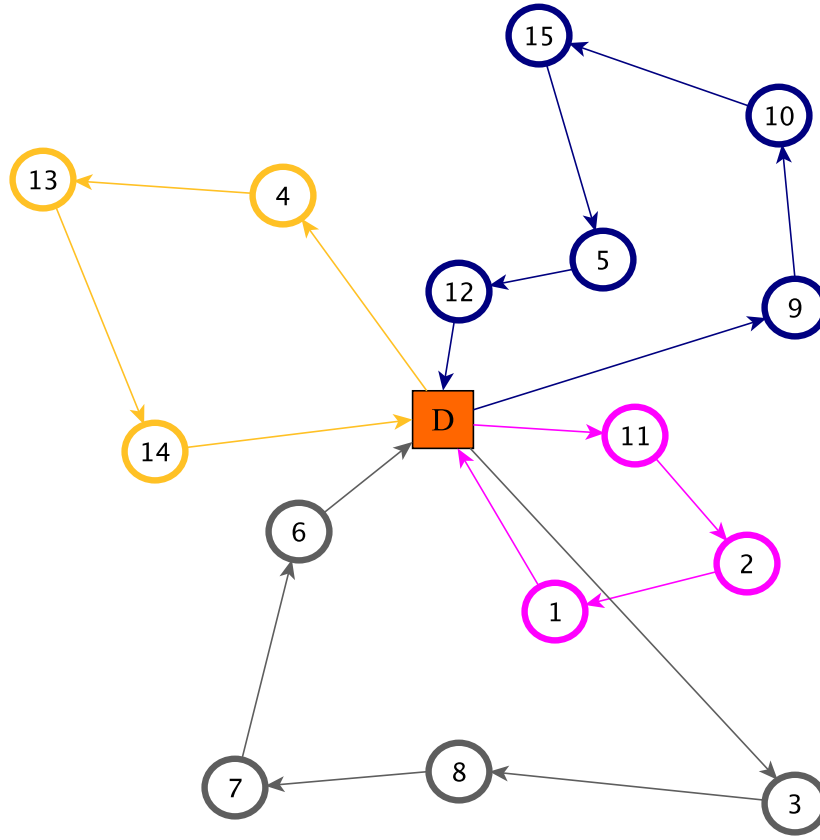


Figura 3.3: Enrutamiento para la Instancia 1 (empaquetamiento factible)

La metodología propuesta mejora la calidad de la solución encontrada por algoritmos previos publicados en la literatura. Los tiempos computacionales son altos debido al uso de una metodología exacta para el problema de enrutamiento de vehículos asociado. Las soluciones obtenidas para el problema integrado 3L-CVRP incluyendo restricciones de empaquetamiento secuencial, dificultan la generación de rutas solución para la distribución de los productos a los clientes. La Figura 3.3 muestra las rutas producidas para la Instancia 1. Nótese que algunas rutas no son envolventes convexas como suelen producirse en soluciones del CVRP tradicional. Además, las rutas generadas muestran con claridad un empeoramiento de la función objetivo para el CVRP, por ejemplo, el traslape de las envolventes que describen el recorrido de los vehículos, dejan en evidencia que las rutas no han sido trazadas con clientes que se encuentran próximos geoméricamente causando un empeoramiento de la minimización de costos de transporte.

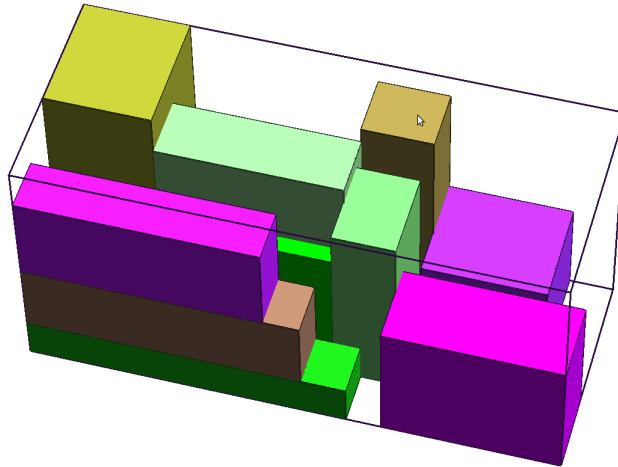


Figura 3.4: Orden de atención para la primera ruta (vehículo 1): 3-8-7-6

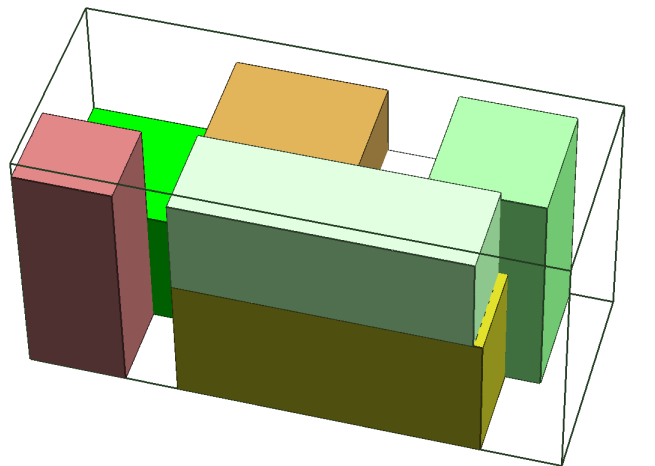


Figura 3.5: Orden de atención para la segunda ruta (vehículo 2): 4-13-14

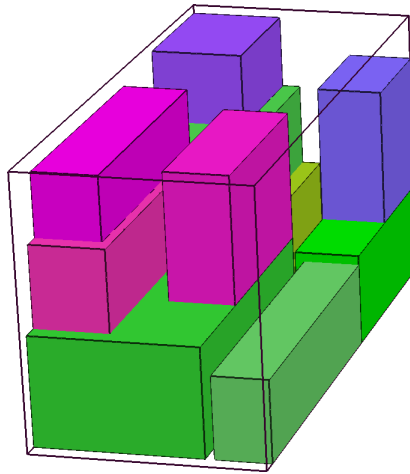


Figura 3.6: Orden de atención para la tercera ruta (vehículo 3): 9-10-15-5-12

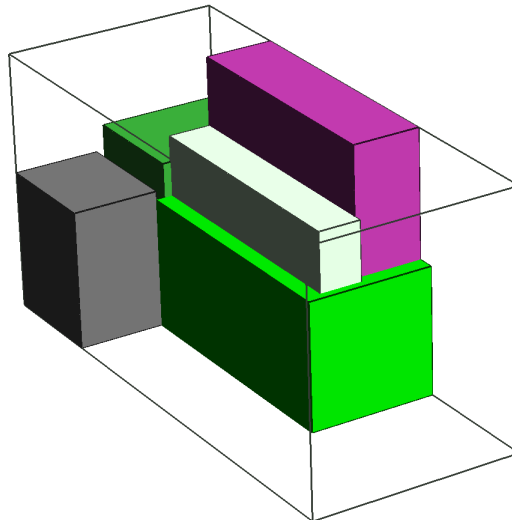


Figura 3.7: Orden de atención para la cuarta ruta (vehículo 4): 11-2-1

El algoritmo propuesto siempre se movió en la infactibilidad, empeorando iterativamente los costos de transporte hasta alcanzar una solución factible y de mínimo costo. En efecto, el CVRP puede considerarse como un límite inferior (lejano) para el problema 3L-CVRP.

Las columnas **Metaheurística** y **Metaheurística (3L-VRP)** de la Tabla 3.1 abren una discusión en torno a las limitantes que tienen las instancias de la literatura especializada para el 3L-CVRP, las cuales, al corresponder a una extensión de las instancias del CVRP, las piezas demandadas presentan una densidad de carga que no tiene sentido en aplicaciones reales. Dado que la demanda de los clientes del CVRP se convierte en el peso de las piezas requeridas por cada cliente en el 3L-CVRP, las dimensiones de las cajas demandadas por un solo cliente pueden variar mucho, y la densidad de estas no tiene sentido en un contexto práctico. Se sugiere entonces observar también el comportamiento ignorando este peso y dejando en manos del volumen ocupado por las cajas el tope de carga que puede recibir cada uno de los vehículos, adicional al enfoque que trae la literatura desde que el problema fue propuesto en (Gendreau et al., 2006).

Para el caso del transporte de un tipo de estructura de las cajas, los resultados obtenidos del enfoque propuesto respecto a las restricciones de empaquetamiento (ver columna **Mateheurística (3L-VRP)** en la Tabla 3.1) están alineados con las conclusiones de Alonso et al. (2014). Efectivamente, las soluciones obtenidas para el conjunto de instancias de pruebas son como mínimo, de la misma calidad que el caso donde todos los tipos de densidad de las cajas para cada cliente son considerados (ver columna **Mateheurística** en la Tabla 3.1).

3.6. Conclusiones y trabajos futuros

Se realizó una revisión de los diferentes trabajos en la literatura que resuelven el problema integrado de enrutamiento de vehículos con restricciones carga de contenedores. La mayoría de trabajos presentan algoritmos aproximados.

En este trabajo, un algoritmo mateheurístico exitoso ha sido propuesto para resolver el 3L-CVRP. La metodología híbrida descompone el 3L-CVRP en dos subproblemas: el

Problema Capacitado de Enrutamiento de Vehículos (CVRP) y el Problema de Carga de Contenedores Tridimensional Secuencial (3D-SLOPP). El algoritmo propuesto combina un algoritmo de *Branch-and-Cut* para resolver el CVRP con un esquema GRASP para encontrar soluciones factibles para el 3D-SLOPP. El algoritmo propuesto ha sido comparado con (Gendreau et al., 2006), (Tarantilis et al., 2009), (Fuellerer et al., 2010), (Ruan et al., 2013), (Bortfeldt, 2012) y (Tao and Wang, 2015) en la librería clásica de instancias propuesta para el 3L-CVRP. Los resultados muestran la efectividad del enfoque propuesto (algunas mejores soluciones son encontradas).

Para trabajos futuros, se consideran otras formulaciones matemáticas (por ejemplo la formulación matemática de tres índices), que puede ser descompuesta explotando los beneficios de la técnica del *Branch-and-Cut* o la generación de columnas. Esta consideración tiene ventajas para resolver el CVRP debido a que provee control de las rutas de manera individual. Adicional a esto, es posible remover fácilmente las restricciones de capacidad y subtour. Sin embargo, es necesario realizar un tratamiento cuidadoso de este modelo, porque implica un crecimiento importante del número de variables frente al modelo empleado en este estudio.

Capítulo 4

Metaheurística híbrida de Búsqueda Tabú Granular y GRASP para el 3L-CVRP

4.1. Introducción

En este capítulo se ofrece una metodología metaheurística híbrida para el 3L-CVRP. El algoritmo utiliza una solución inicial obtenida por un algoritmo Clarke & Wright considerando restricciones de empaquetamiento, controladas por un esquema GRASP, durante el proceso de búsqueda.

4.2. Estructura del Algoritmo Propuesto

El algoritmo híbrido propuesto consta de tres partes: 1) construcción de una solución inicial aplicando un procedimiento modificado propuesto por [Clarke and Wright \(1964\)](#) (Algoritmo 2), 2) un Algoritmo de Búsqueda Tabú Granular GTS (por sus siglas en inglés *Granular Tabu Search*) para mejorar las distancias recorridas en las rutas durante el proceso de búsqueda

(Algoritmo 4) y 3) un algoritmo GRASP para ubicar la demanda acumulada de cada una de las rutas construidas (Algoritmo 3). En las siguientes subsecciones, se detallan las partes de la metodología.

4.2.1. Solución Inicial

La solución inicial S_0 es construida utilizando un procedimiento Clarke & Wright modificado introducido en [Clarke and Wright \(1964\)](#). Esta estrategia es aplicada para obtener soluciones iniciales de buena calidad en tiempos computacionales cortos. esta variación del Clarke & Wright permite considerar las restricciones de carga tridimensionales. El procedimiento considera la reasignación de los vehículos a grupos de clientes, de acuerdo a la redistribución de la demanda total. Cuando un grupo de clientes (ruta) es generado, la factibilidad por empaquetamiento es determinada por el Algoritmo 3. El Algoritmo 2 presenta el pseudocódigo del proceso descrito en esta subsección.

Algorithm 2 Modified Savings Algorithm

Data: Customers, Distances, Vehicles

Result: Initial solution S_0

```

1 Calculate savings list;
2 Sort decreasingly the savings list;
3 for all saving in savings list do
4   Calculate demand weight and three-dimensional packing feasibility for the customers group
   demand (Algorithm 3);
5   if packing constraint is fulfilled then
6     Apply saving;
7   else
8     Ignore saving;
9   end
10 end
11 return Initial solution  $S_0$ , considering fulfilling packing constraints but allowing capacity
unfeasibility, which will be repaired in the main algorithm
```

Esta forma de obtener la solución inicial se puede clasificar como golosa, porque todas las rutas son construidas a través de una búsqueda continua de una solución local mínima. Esta estrategia permite obtener más rutas que vehículos disponibles en el depósito, es decir, la restricción que controla el número máximo de rutas es relajada durante el procedimiento. Las rutas excedentes son eliminadas durante la fase de la Búsqueda Tabú Granular.

4.2.2. Algoritmo GRASP

El algoritmo GRASP ha sido propuesto por [Feo and Resende \(1989\)](#) para resolver problemas combinatoriales duros. El GRASP es un proceso iterativo que combina una fase constructiva con una fase de mejora. En la fase constructiva, una solución es generada paso a paso, agregando elementos a una solución parcial. La fase constructiva es iterativa, golosa, aleatorizada y adaptativa, mientras que en la fase de mejora, algunos movimientos son incluidos en la estructura iterativa con el fin de intensificar y diversificar la búsqueda de la solución. Diferentes estudios demuestran su calidad y robustez, para una mejor comprensión, el lector es invitado a revisar [Resende and Ribeiro \(2003\)](#).

Algoritmo Constructivo

El algoritmo constructivo está inspirado en el procedimiento propuesto por [Parreno et al. \(2010\)](#) para el problema de carga de contenedores. El algoritmo está basado en espacios maximales. En la medida en la que una caja seleccionada es empacada en un nuevo espacio, tres nuevos espacios maximales son producidos. El algoritmo constructivo utiliza una lista actualizada de espacios maximales, y una lista de cajas para el cliente actual, las cuales están en proceso de ser empacadas. La parte más difícil para este caso, es la gestión de los espacios vacíos cuando todas las cajas han sido empacadas para cierto cliente. En dicho caso, antes de empezar a empacar las cajas para el siguiente cliente, algunos espacios tienen que ser removidos de la lista y otros tienen que ser actualizados, con el fin de satisfacer las restricciones *multi-drop* o de carga secuencial. Esta idea ha sido introducida por [Parreno et al. \(2010\)](#) para el problema de empaquetamiento tridimensional.

■ *Paso 0: Inicialización*

FS = lista de espacios maximales vacíos creada cuando se empacan las cajas. Inicialmente, FS es simplemente el contenedor vacío.

$B = \{B_1, B_2, \dots, B_n\}$ el conjunto de cajas pendientes para empacar, ordenadas por cliente.

■ *Paso 1: Eligiendo el espacio maximal de S*

Se considera el listado FS de espacios maximales, los cuales son los paralelepípedos vacíos más grandes disponibles para ser llenados con cajas. Se utilizan dos estrategias alternativas para seleccionar el espacio maximal. Se alterna entre seleccionar el espacio maximal con la distancia mínima al respaldo del contenedor, y seleccionar el espacio maximal con la coordenada z mayor. En efecto, esta estrategia intenta llenar el fondo del contenedor primero o acomodar las cajas formando pilas.

■ *Paso 2: Seleccionando las cajas que serán empacadas*

Una vez que un espacio maximal FS ha sido seleccionado, consideramos las cajas restantes del cliente actual que ajustan en FS con el fin de seleccionar la caja que debe ser empacada. Si hay muchas cajas con las mismas dimensiones, se estudia la posibilidad de empacarlas como un bloque, es decir, empacar muchas de esas cajas dispuestas en un arreglo rectangular con muchas filas y columnas. Dos criterios han sido considerados para seleccionar la configuración de las cajas: Mejor-Volumen *Best-Volume* y Mejor-Ajuste *Best-Fit*. En el primero, la caja o el bloque de cajas produciendo el mayor incremento en el volumen ocupado por las mismas es seleccionado. En el segundo, la caja o bloque de cajas que presentan el mejor ajuste en el espacio maximal es considerada. Se computan las distancias desde cada lado de la caja o bloque a cada lado del espacio maximal, y se ponen esos espacios en un vector en orden no-decreciente. La caja o bloque de cajas es seleccionado utilizando orden lexicográfico.

Si el criterio de visibilidad es utilizado, la esquina, dentro de la cual la caja o el bloque es empacado, es el espacio maximal con la distancia más corta al origen del contenedor. En los otros dos casos, el algoritmo revisa la posición y la ubicación de la caja o bloque

de cajas con el fin de hacerla alcanzable, siempre intentando ponerla tan cerca del fondo como sea posible, buscando maximizar el espacio restante utilizable del contenedor.

■ *Paso 3: Actualizando la lista FS para el cliente actual*

A menos que la caja o bloque de cajas ajuste exactamente dentro del espacio FS , empacarlo produce nuevos espacios maximales vacíos que son reemplazados en la lista FS . Además, mientras los espacios maximales son no disjuntos, la caja o bloque de cajas que están siendo empacadas pueden ser intersectadas por otro espacio maximal, el cual tendrá que reducirse. Una vez los nuevos espacios han sido añadidos y algunos de los existentes modificados, se revisa la lista y se eliminan posibles inclusiones. El listado B también es actualizado y los espacios maximales que no pueden recibir ninguna de las cajas pendiente por ser empacadas son eliminados de FS . Si $FS = \emptyset$ o $B = \emptyset$, el procedimiento termina. De lo contrario, si no se ha completado el empaquetado de las cajas para el cliente actual, se retorna al Paso 1.

■ *Paso 4: Actualizando el listado FS para un nuevo cliente*

Se tiene que actualizar el listado de espacios maximales para el siguiente cliente. Para eso, el enfoque propuesto elimina del listado todos los espacios maximales que no son completamente visibles desde el frente del contenedor. Luego, los espacios maximales que tienen algunas secciones visibles y otras no visibles son actualizados.

Búsqueda Local (Mejoras)

El enfoque propuesto considera dos tipos de mejora o procedimientos de búsqueda local. El primero intenta mejorar el patrón de empaquetamiento final, mientras que el segundo concentra su esfuerzo computacional en obtener un mejor patrón de empaquetamiento para cada cliente.

- La primera mejora consiste en la eliminación del último ($k\%$) de cajas empacadas en el patrón final de empaquetamiento. (Por ejemplo el 50% final). El valor k es seleccionado aleatoriamente en el intervalo $[30, 90]$. Las cajas removidas, más las cajas

que se encuentran por fuera de la solución, son empacadas nuevamente utilizando un procedimiento constructivo determinístico. Las funciones objetivo del algoritmo constructivo son *Mejor-Volumen* y *Mejor-Ajuste*. Esta nueva solución es considerada solamente si el volumen total de las cajas empacadas aumenta.

- En la segunda mejora, la metodología aplica el procedimiento previamente descrito a la solución parcial obtenida después de realizar el empaquetamiento de un cliente. Específicamente, esta nueva solución es considerada solamente si el área ocupada es reducida.

En este estudio se ha desarrollado un algoritmo constructivo aleatorizado, basado en espacios maximales que construye soluciones totalmente factibles. Adicionalmente, se propone una fase de búsqueda local con varios movimientos de mejora como: construcción de columnas, bloques y filas, y compresión de soluciones por clientes. Las dos fases son combinadas en un esquema metaheurístico reactivo tipo GRASP. El algoritmo 3 muestra cómo el GRASP valida las restricciones de empaquetamiento y produce patrones de empaquetamiento *multi-drop* si los ítems pueden ser cargados dentro del vehículo.

4.2.3. Búsqueda Tabú en Espacio Granular & Algoritmo de Validación 3L tipo GRASP

El abordaje propuesto (GTS-GRASP) utiliza el concepto de espacio de búsqueda granular propuesto por [Toth and Vigo. \(2003\)](#), el cual está basado en la utilización de un grafo disperso conteniendo las aristas incidentes a los depósitos, las aristas perteneciendo a la mejor solución encontrada hasta el momento, y las aristas cuyo costo es menor al umbral de granularidad $\vartheta = \beta \bar{z}$, donde \bar{z} es el costo promedio de las aristas en la mejor solución encontrada, y β es un factor de dispersión que es actualizado dinámicamente durante la búsqueda. La modificación del valor de β permite al algoritmo alternar entre etapas de intensificación (valores pequeños de β) y diversificación (valores grandes de β). La meta principal del Búsqueda Tabú Granular es obtener soluciones de alta calidad manteniendo las principales características del Búsqueda Tabú en cortos tiempos computacionales ([Escobar et al. \(2013\)](#), [Escobar et al. \(2014b\)](#)),

Algorithm 3 GRASP_3L_Validation

Data: Lists of Spaces FS , List of Boxes B_c of each Customer c and List of Packing Patterns P

Result: Feasible or Unfeasible packing.

```
1 Initialize  $FS = OriginalContainerSize, B = ListOfBoxesOfEachCustomer, P = \emptyset$ 
2 for  $l \leftarrow 1 \dots IterationsNumber$  do
3   while  $FS \neq \emptyset$  do
4     while  $B_c \neq \emptyset$  do
5       Space  $FS_i = SelectSpaceByCriteria(FS)$ ;
6       List  $C_S = GenerateBlockList(FS_i, B)$ ;
7       List  $RCL = BuildRCL(C_S, \delta)$ ;
8       Layer  $C = SelectBlockRandomly(RCL)$ ;
9       Packing Pattern  $P = LocateBlock-Space(C, FS_i, P)$ ;
10      List  $FS = UpdateListOfMaximalSpaces(C, FS)$ ;
11      List  $B = UpdateListOfRemainingBoxes(C, B)$ ;
12    end
13    List  $FS = UpdateListOfSpacesByCostumerChanging(FS, Multi - dropType)$ ;
14  end
15 end
16 if  $B \neq \emptyset$  then
17   return Feasible;
18 end
19 return Unfeasible;
```

[Escobar et al. \(2014a\)](#)).

El Algorithm GTS-GRASP permite soluciones infactibles con respecto a la capacidad de los vehículos. Dada una solución S compuesta por un conjunto de z rutas (r_1, \dots, r_z) , y cada ruta r_l , donde $l \in \{1, \dots, z\}$, es denotada por $(v_0, v_1, v_2, \dots, v_0)$. Adicionalmente, denotando $v \in r_l$ como un cliente v perteneciendo a la ruta r_l , y $(u, v) \in r_l$ siendo una arista donde u y v son dos vértices consecutivos de la ruta r_l . Se asigna a la solución S un valor de la función objetivo $F_1(S) = \sum_{l=1}^z \sum_{(u,v) \in r_l} c_{uv}$. Si la solución S es infactible, se le asigna a S un valor de función objetivo $F_2(S) = F_1(S) + P_q(S)$, donde $P_q(S)$ es un término de penalidad obtenido multiplicando el global sobre la capacidad de vehículos de la solución S , por un factor de penalidad actualizado dinámicamente α_q . En particular $\alpha_q = \rho_q \times F_1(S_0)$, donde ρ_q es un parámetro calculado durante la búsqueda. Nótese que si la solución actual S es factible, $F_1(S) = F_2(S)$.

Si durante la búsqueda, se encuentran soluciones infactibles con respecto a la capacidad de vehículos para un número dado N_{fact} de iteraciones, el valor de ρ_q es fijado en $\min\{\rho_{max}, \rho_q \times \delta_{inc}\}$, donde $\delta_{inc} > 1$. De lo contrario, si cualquier solución factible ha sido encontrada para un número dado N_{fact} de iteraciones, el valor de ρ_q es establecido en $\max\{\rho_{min}, \rho_q \times \delta_{red}\}$, donde $\delta_{red} < 1$. N_{fact} , ρ_{min} , ρ_{max} , δ_{inc} , and δ_{red} son parámetros dados.

El algoritmo explora el espacio de soluciones moviéndose en cada iteración, desde la solución S_i hasta la mejor solución en el vecindario $N_k(S_i)$, incluso si esta es infactible. El movimiento seleccionado es declarado como *tabu*. El *tabu tenure* es especificado como un valor entero aleatorio en el rango $[t_{min}, t_{max}]$, donde t_{min} y t_{max} son parámetros dados. La estrategia de diversificación está basada en la idea granular propuesta por [Toth and Vigo. \(2003\)](#). Inicialmente, el factor de dispersión β es establecido en un valor pequeño β_0 . Si S_i es infactible después de N_β iteraciones, el factor de dispersión β es aumentado a un valor β_d . Luego, un nuevo grafo es calculado, y N_{change} iteraciones son realizadas desde la mejor solución factible encontrada (S_{best}). Finalmente, si S_i es factible después de N_{change} iteraciones, el factor de dispersión es reiniciado a su valor original β_0 y la búsqueda continua. β_0 , β_d , N_β y N_{change} son parámetros dados.

El algoritmo metaheurístico híbrido propuesto (GTS-GRASP) utiliza movimientos intra-ruta e inter-ruta correspondientes a los siguientes vecindarios:

- **Swap:** Dos clientes (en la misma ruta o en rutas diferentes) intercambian sus posiciones.
- **Double-Swap:** Este movimiento es una extensión del movimiento *Swap* obtenido al considerar dos pares compuestos de clientes consecutivos. La arista conectando cada par de clientes es conservada.
- **Two-Opt:** La estrategia 2-opt ampliamente utilizada, es empleado en este estudio para movimientos tanto intra-ruta como inter-ruta.
- **Shift(1,0):** Este operador es realmente útil para reparar el número de rutas de la solución inicial S_0 . En efecto, este operador permite reparar la infactibilidad S_0 respecto al número de rutas (solución producida por el algoritmo constructivo de ahorros modificado propuesto en este trabajo) decrementando el número de rutas al reubicar algunos clientes (transfiriendo algunos clientes entre rutas). El Shift(1,0) siempre reubica un cliente de una ruta i a otra ruta j .

Algorithm 4 GTS-GRASP Algorithm

Data: Lists C (Customers), D (Demand of Customers)

Result: Routes and packing patterns

```
1  $S_0 \leftarrow \text{Clark\&Wright-GRASP}(C, D)$ 
2 Start GTS parameters
3  $S_{best} \leftarrow S_0$ 
4 for  $i \leftarrow 1 \dots \text{GTS\_Number\_of\_Iterations}$  do
5     if  $i == 1$  then
6          $S_i \leftarrow S_0$ 
7     end
8      $\text{listOfMoves} \leftarrow \emptyset$ 
9      $\text{listOfMoves} \leftarrow \text{listOfMoves} \cup \text{2opt}(S_i) \cup$ 
10          $\text{insertion}(S_i) \cup \text{doubleInsertion}(S_i) \cup$ 
11          $\text{swap}(S_i) \cup \text{doubleSwap}(S_i) \cup \text{shift}(S_i)$ 
12      $\text{listOfMoves.ascendingSort}(\text{MovementCost})$ 
13     for  $j \leftarrow 1 \dots \text{listOfMoves.size}$  do
14         if  $\text{GRASP\_3L\_Validation}(\text{movement}_j)$  then
15              $S'_i \leftarrow \text{applyMovement}(\text{movement}_j)$ 
16         end
17         break
18     end
19     if  $f(S_i) > f(S'_i)$  then
20          $S_i \leftarrow S'_i$ 
21          $S_{best} \leftarrow S_i$ 
22     end
23     if  $N_{shake}$  is reached then
24          $\text{Perturbation\_Function}(S_{best}, S_i, \text{Seed\_Param})$ 
25          $\text{Seed\_Param}++$ 
26         Number of iterations without updating  $S_{best} = 0$ 
27     end
28     Update GTS parameters
29 end
30 return  $S_{best}$ 
```

La metodología desarrollada es presentada en el Algoritmo 4. El algoritmo es repetido hasta que $N_{iterations}$ son realizadas. El *GRASP-3L Validation* sera aplicado exclusivamente a los movimientos más promisorios del listado de movimientos *listOfMoves* (prioridad por menor costo) obtenido por los operadores, pero la revisión de factibilidad de empaquetamiento es realizada hasta que un movimiento 3L factible es encontrado en *listOfMoves*.

Los movimientos más promisorios realizados por los operadores, presentes en *listOfMoves*, son considerados teniendo en cuenta solamente las restricciones correspondientes al problema de enrutamiento de vehículos asociado. Es por esto que algunas soluciones pueden ser infactibles en términos de restricciones de empaquetamiento tridimensional. Esta situación causa que buenas soluciones de enrutamiento puedan ser ignoradas del listado de movimientos *listOfMoves* a causa de su infactibilidad en términos de empaquetamiento tridimensional. Adicionalmente, la primera solución 3L y *multi-drop* factible, usualmente tiene peor calidad con respecto a la mejor solución encontrada hasta el momento S_{best} para el problema considerado, llevando a un considerable desperdicio de iteraciones y de esfuerzo computacional.

Con el fin de evitar óptimos locales, un procedimiento de perturbación es aplicado para producir soluciones a partir del mejor patrón de enrutamiento encontrado hasta el momento, teniendo una mayor probabilidad de obtener soluciones para el problema completo (3L-CVRP). El esquema de perturbación es aplicado después de completar un número N_{shake} de iteraciones sin actualizar la mejor solución S_{best} encontrada hasta el momento. En la etapa de parametrización, se encontró que el número apropiado de iteraciones permitido sin actualizaciones de S_{best} es 50.

El procedimiento de perturbación, a diferencia del procedimiento de reparación de la solución actual S_i , permite explorar muchas soluciones obtenidas de la mejor solución encontrada S_{best} controlada aleatoriamente. Este procedimiento permite explorar nuevas y diversas áreas promisorias de búsqueda. S'_{best} es obtenida inicialmente a partir de la mejor solución encontrada hasta el momento, la cual es modificada aleatoriamente de la siguiente manera: 1) dos rutas son seleccionadas aleatoriamente (routes r_l y t_l); 2) selección aleatoria de un

cliente (p) de la ruta r_l y un cliente (q) de la ruta t_l ; 3) el cliente p es removido de la ruta r_l e insertado dentro de la ruta t_l en la posición de q ; 4) el cliente q es removido de la ruta t_l e insertada en una ruta u_l seleccionada aleatoriamente, donde $r_l \neq u_l$. Nótese que el cliente n es insertado en el mejor lugar (mejor función objetivo) de la ruta u_l . El procedimiento de perturbación es presentado en la Figura 4.1, donde (a) muestra la selección de las rutas y los clientes, y (b) muestra el movimiento realizado. Para el procedimiento de perturbación, las restricciones de empaquetamiento son también validadas. Por lo tanto, el procedimiento es repetido hasta que una nueva solución factible S'_{best} , derivada de la solución S_{best} es obtenida, es decir, $S_i \leftarrow S'_{best}$. En el Algoritmo 5, el procedimiento de perturbación es detallado.

Algorithm 5 Perturbation.Function

Data: S_{best} , S_i , $Seed_Parameter$

Result: Perturbed S_i

```

1 Perturbation_Flag  $\leftarrow$  false;
2  $S'_{best} \leftarrow S_{best}$ ;
3 while  $\neg$ Perturbation_Flag do
4     Select randomly two different routes  $r_l$  and  $t_l$  from  $S'_{best}$ ;
5     Select a random customer  $p$  from route  $r_l$ ;
6     Select a random customer  $q$  from route  $t_l$ ;
7     Move customer  $p$  from route  $r_l$  to route  $t_l$  replacing the position of customer  $q$ ;
8     Move customer  $q$  to a route  $u_l$  with  $r_l \neq u_l$ ;
9     if packing constraints are fulfilled then
10         Perturbation_Flag  $\leftarrow$  true;
11          $S_i \leftarrow S'_{best}$ ;
12     else
13          $S'_{best} \leftarrow S_{best}$ ;
14     end
15 end
16 return  $S_i$ 

```

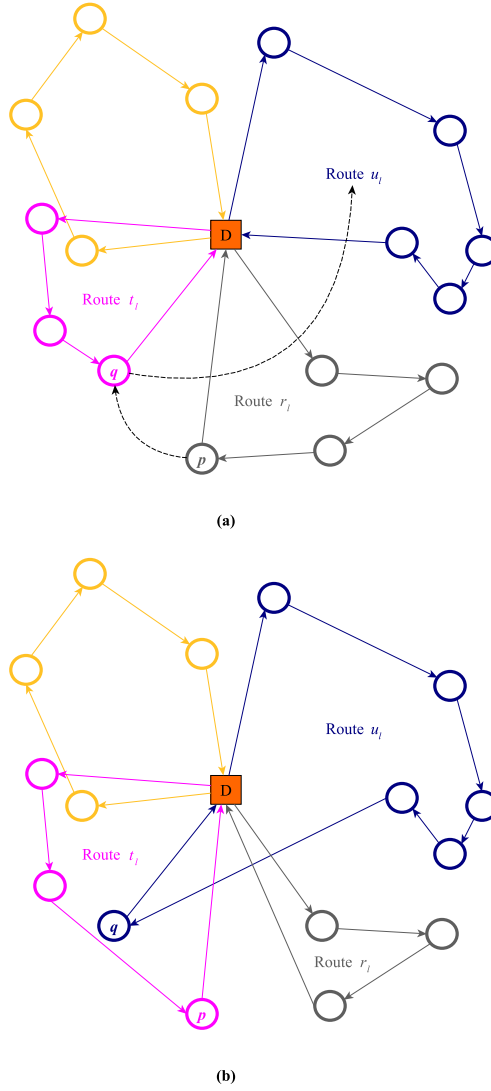


Figura 4.1: Perturbación de la mejor solución encontrada hasta alguna iteración de la metodología S_{best} , para obtener S'_{best} como nuevo punto de búsqueda S_i

Tabla 4.1: Parametrización para el Algoritmo GTS-GRASP

Parámetro	Descripción	Valor
β_0	Valor inicial del factor de dispersión	2
ρ_{min}, ρ_{max}	Mínimo y máximo valor para penalidad por capacidad	$[8, 8 \times F_1(S)]$
β_d	Delta del factor de dispersión	4
$\delta_{inc}, \delta_{red}$	Factores de incremento y reducción para la, penalidad por capacidad	$1, 1, \frac{1}{1,1}$
N_{shake}	Número de iteraciones permitidas sin actualización de S_{best}	50
$N_{iterations}$	Número total de iteraciones para el Algoritmo GTS-GRASP	$100 \times n$
$Tenure$	Tabu Tenure	50

4.3. Experimentos Computacionales

El algoritmo propuesto (GTS-GRASP) ha sido implementado en C++ y los experimentos han sido desarrollados con un procesador *Intel(R) Core(TM) i7* 2.93 GHz, 16 Gb RAM, bajo sistema operativo Ubuntu 14.04 de 64 bits. La efectividad del algoritmo propuesto ha sido verificada con el conjunto de instancias de prueba propuestas por [Gendreau et al. \(2006\)](#). Este conjunto consiste en 27 instancias, cuyo número de clientes n varía entre 15 y 100 clientes.

4.3.1. Parametrización

Un gran número de pruebas se realizaron con el fin de obtener un conjunto de parámetros para ser utilizados en el conjunto completo de instancias de prueba. En la Tabla 4.1 se muestra el mejor conjunto de parámetros obtenido para todas las instancias.

En la Tabla 4.2 los resultados obtenidos por la metodología propuesta son presentados. De manera complementaria, en esta misma Tabla, se presenta información de las publicaciones previas a este trabajo, para realizar una comparación completa del Algoritmo GTS-GRASP con los mejores resultados computacionales disponibles en la literatura.

Tabla 4.2: Comportamiento del algoritmo y comparación con los mejores algoritmos publicados hasta el momento del estudio

Instancia	Nodos	Ítems	Vehículos	Gendreau et al. (2006)		Tarantilis et al. (2009)		Fuellerer et al. (2010)		Bortfeldt and Homerger (2013)		Ruan et al. (2013)		Tao and Wang (2015)		GTS-GRASP				
				Solución	Tiempo	Solución	Tiempo	Mejor	Promedio	Tiempo	Mejor	Promedio	Tiempo	Solución	Tiempo	Mejor	Promedio	Tiempo	Solución	Tiempo Total
1	15	32	4	316.32	1800	321.47	13.2	304.13	305.35	12	302.02	302.02	72.3	303.21	98.85	302.02	302.02	53.5	300.7	107
2	15	26	5	350.58	1800	334.96	11.5	334.96	334.96	0.6	334.96	334.96	0.9	334.96	4.55	334.96	334.96	6.3	340.55	7
3	20	37	4	447.73	1800	430.95	540.6	399.68	409.79	121.8	392.63	401.44	182	398.05	93.86	381.37	381.37	116.2	404.03	154
4	20	36	6	448.48	1800	458.04	323.5	440.68	440.68	5.4	437.19	437.54	16.1	440.68	46.75	437.19	437.19	14	430.89	8
5	21	45	6	464.24	1800	465.79	99.6	450.93	453.19	30.9	443.61	451.03	182.6	452.56	63.98	436.48	436.79	149.3	492.24	456
6	21	40	6	504.46	1800	507.96	1212.4	498.32	501.47	18.4	498.16	498.38	23.6	498.56	196.9	498.32	498.32	31.6	498.32	17
7	22	46	6	831.66	1800	796.61	364.8	792.13	797.47	67.4	769.68	772.49	133.1	790.23	317.02	767.46	768.94	84.9	789.78	335
8	22	43	6	871.77	1800	880.93	230	820.67	820.67	78.6	810.89	821.35	139.1	820.67	98.9	804.75	805.77	120.4	875.08	496
9	25	50	8	666.1	1800	642.22	982.2	635.5	635.5	16.3	630.13	645.81	24.3	635.5	353.07	630.13	631.68	13.7	639.26	78
10	29	62	8	911.16	3600	884.74	1308.4	840.75	841.12	246.7	820.35	827.29	175.1	836.21	410.9	822.86	828.99	258.6	829.23	2063
11	29	58	8	819.36	3600	873.43	522.5	818.87	821.04	199.8	803.61	815.62	136.4	825.75	197.76	772.85	780.61	278.9	762.51	1370
12	30	63	9	651.58	3600	624.24	294.6	626.37	629.07	48.2	614.59	630.46	14	626.59	89.47	614.59	614.6	145.8	641.3	62
13	32	61	8	2928.34	3600	2799.74	2193.1	2739.8	2739.8	308.8	2645.95	2694.81	268.4	2739.8	319.78	2608.68	2636.85	369.4	2759.12	2017
14	32	72	9	1559.64	3600	1504.44	4581.3	1466.84	1472.26	642.8	1368.42	1413.59	311.6	1469.38	268.39	1391.56	1398.77	588.1	1482.88	2000
15	32	68	9	1452.34	3600	1415.42	2528.3	1367.58	1405.48	656.8	1341.14	1355.5	311.5	1369.69	356.55	1345.72	1352.76	615.9	1374.22	2000
16	35	63	11	707.85	3600	698.61	4256.5	698.92	698.92	14.8	698.61	705.05	3.4	703.15	431.74	698.92	698.92	4	703.38	11
17	40	79	14	920.87	3600	872.79	2096	868.59	870.33	14.9	866.4	917.96	2.5	872.05	374.84	866.4	866.4	9.5	871.63	23
18	44	94	11	1400.52	3600	1296.59	2275.2	1255.64	1261.07	2209.8	1207.72	1228.98	309.5	1250.86	325.74	1220.3	1228.47	1634.1	1415.46	2000
19	50	99	12	871.29	7200	818.68	2509	777.18	781.29	623.6	741.74	753.87	416.5	780.37	1374.84	760.97	763.09	718.4	838.45	2000
20	71	147	18	732.12	7200	641.57	1940.9	604.28	611.26	3901	587.95	596.42	427	605.59	1336.97	589.66	590.99	2941.8	634.27	2000
21	75	155	17	1275.2	7200	1159.72	2823.4	1110.09	1124.55	5180.6	1090.22	1107	443.4	1119.45	1247.86	1086.17	1096.53	2301.4	1147.64	2000
22	75	146	18	1277.94	7200	1245.35	2685.6	1194.18	1197.43	2290.3	1147.8	1171.49	423.5	1167.28	1294.57	1150.55	1155.81	1241.8	1218.54	2000
23	75	150	17	1258.16	7200	1231.92	4659.1	1158.51	1171.77	3727.6	1130.54	1135.46	425.8	1171.77	1105.74	1127.92	1130.08	1924.9	1133.71	2000
24	75	143	16	1307.09	7200	1201.96	4854.1	1136.8	1148.7	1791.5	1116.13	1128.82	411.1	1136.27	2001.05	1114.1	1122.8	2526.8	1189.25	2000
25	100	193	22	1570.72	7200	1457.46	5725.8	1429.64	1436.32	8817.1	1407.36	1428.8	453	1426.34	1458.8	1415.75	1417.09	4536.2	1464.03	2000
26	100	199	26	1847.95	7200	1711.93	6283.1	1611.78	1616.99	6904.3	1600.35	1625.31	430.6	1585.46	3354.72	1602.06	1605.11	3017.7	1664.84	2000
27	100	198	23	1747.52	7200	1646.44	9915.7	1560.7	1573.5	10483.9	1529.86	1550.85	435	1562.18	3140.18	1535.83	1538.1	6025.7	1647.88	2000
Promedio (Total)				1042.26		997.18			966.67			953.79		960.10			941.59		983.30	
Relativo				GAP %	-5.7	-1.4			1.7			3.1		2.4			4.4			
Promedio				Tiempo (s)	4200		2416			1793			229		754			1101		1156
CPU				Pentium IV	3.0 GHz	Pentium IV	2.8 GHz	Pentium IV	3.2 GHz		Core2Duo	E8500	3.16GHz	Pentium IV	2.3 GHz	Pentium IV	2.67 GHz	i7-2600S	2.8GHz	
Single	Thread	Rating		653		635		699			1321			564		608		1797		

4.3.2. Análisis Comparativo

El algoritmo propuesto ha sido comparado con los algoritmos más efectivos propuestos para el 3L-CVRP hasta el momento del estudio: TSTS12 [Bortfeldt \(2012\)](#), ACO10 [Fuellerer et al. \(2010\)](#), TSA06 [Gendreau et al. \(2006\)](#) y TSGLS09 [Tarantilis et al. \(2009\)](#) en el conjunto de instancias de *benchmarking* propuestas en [Gendreau et al. \(2006\)](#).

Para cada instancia, el algoritmo propuesto GTS-GRASP ha sido ejecutado una sola vez, de esta forma, la mejor solución encontrada en cada instancia es reportada. Los resultados reportados para los algoritmos TSA06 y TSGL09 corresponden a una sola ejecución de los mismos. Los resultados reportados para los algoritmos ACO10 y TSTS12 corresponden, para cada instancia, al costo promedio encontrado y el mejor costo encontrado considerando 10 ejecuciones de cada algoritmo. Los resultados son reportados tal y como aparecen en las publicaciones consideradas.

Nótese que el enfoque propuesto tiene la capacidad de obtener resultados de buena calidad. De hecho, el GAP promedio de 4.24 % respecto a la mejor solución conocida BKS (por las siglas en inglés de *Best Known Solution*) reportada en la literatura, es alcanzada. Adicional al comportamiento para todas las instancias, cuatro nuevos BKS han sido alcanzados, tres de ellos son resaltados en la Tabla 4.2 obtenidos con los parámetros presentados en la subsección previa (4.3.1). El cuarto BKS fue encontrado durante la etapa de parametrización para la instancia número 23. Los patrones de enrutamiento y empaquetamiento para el BKS correspondiente a la instancia número 1 son presentados en las Figuras 4.2, 4.3, 4.4, 4.5 y 4.6.

Durante la etapa de parametrización, el rendimiento del algoritmo fue notorio para algunas instancias. En efecto, el enfoque propuesto es capaz de obtener la mejor solución conocida para la instancia 23, la cual considera 75 nodos, 150 ítems (cajas) y 17 vehículos. El mejor patrón de enrutamiento es presentado en la Figura 4.7, así como en la Figura 4.8 se presenta la visualización de los patrones de empaquetamiento correspondientes a las rutas que componen el BKS de la instancia 23.

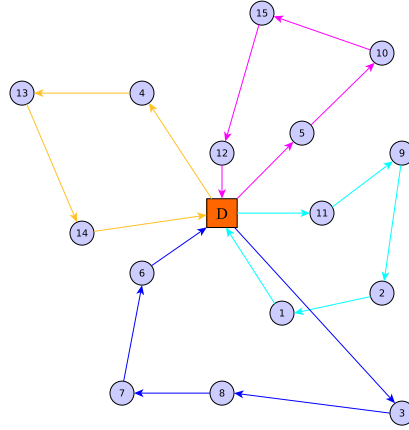


Figura 4.2: Patrón de enrutamiento obtenido por el algoritmo GTS-GRASP para la instancia 1. BKS = 302.02, GTS-GRASP = **300.67**

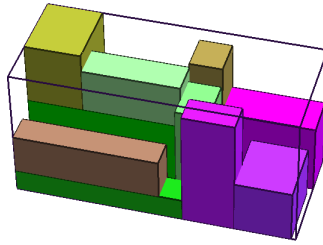


Figura 4.3: Patrón de empaquetamiento para la ruta: 3-8-7-6

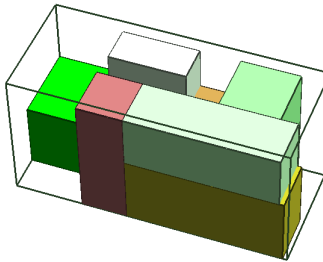


Figura 4.4: Patrón de empaquetamiento para la ruta: 4-13-14

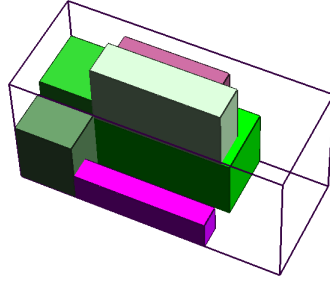


Figura 4.5: Patrón de empaquetamiento para la ruta: 11-9-2-1

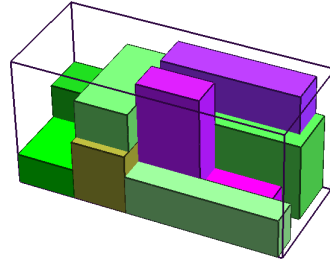


Figura 4.6: Patrón de empaquetamiento para la ruta: 5-10-15-12

4.4. Conclusiones e investigaciones futuras

Se propuso una metodología metaheurística híbrida para el 3L-CVRP denominada GTS-GRASP en este documento. Específicamente, un enfoque tipo Búsqueda Tabú Granular es utilizado para resolver el Problema de Enrutamiento de Vehículos, mientras que un esquema tipo GRASP resuelve el Problema de Carga de Contenedores correspondiente para cada una de las rutas. El enfoque propuesto produce un algoritmo muy competitivo que alcanza la mejora de cuatro de las mejores soluciones conocidas, para instancias de pequeño, mediano y gran porte.

Para trabajos futuros, una buena forma de decrementar los tiempos computacionales de la metodología propuesta, es mejorar el procesamiento de la lista de movimientos utilizando estrategias computacionales especializadas (como paralelización), con el fin de aliviar el cuello de botella que se presenta en las validaciones de empaquetamiento sobre las rutas que se van generando en el proceso. Por otro lado, la calidad del algoritmo también podría mejorarse pasando el GTS a una arquitectura paralela que tenga la capacidad de correr de manera

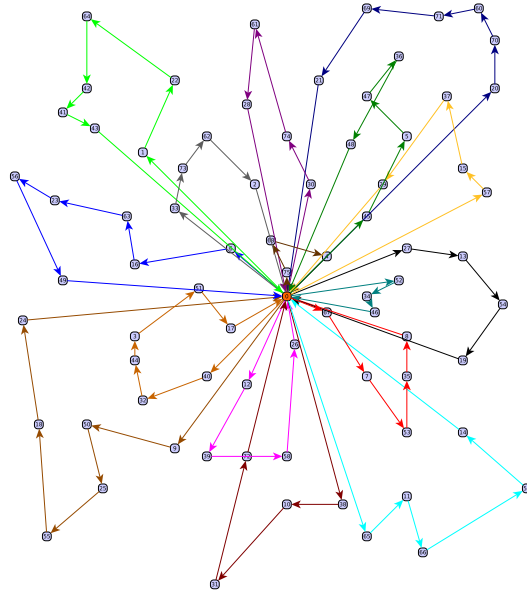


Figura 4.7: Patrón de enrutamiento obtenido por el Algoritmo GTS-GRASP para la instance 23. $BKS = 1130.54$, $GTS-GRASP = \mathbf{1122.48}$

simultánea varias trayectorias tabú con diferentes conjuntos de parámetros.

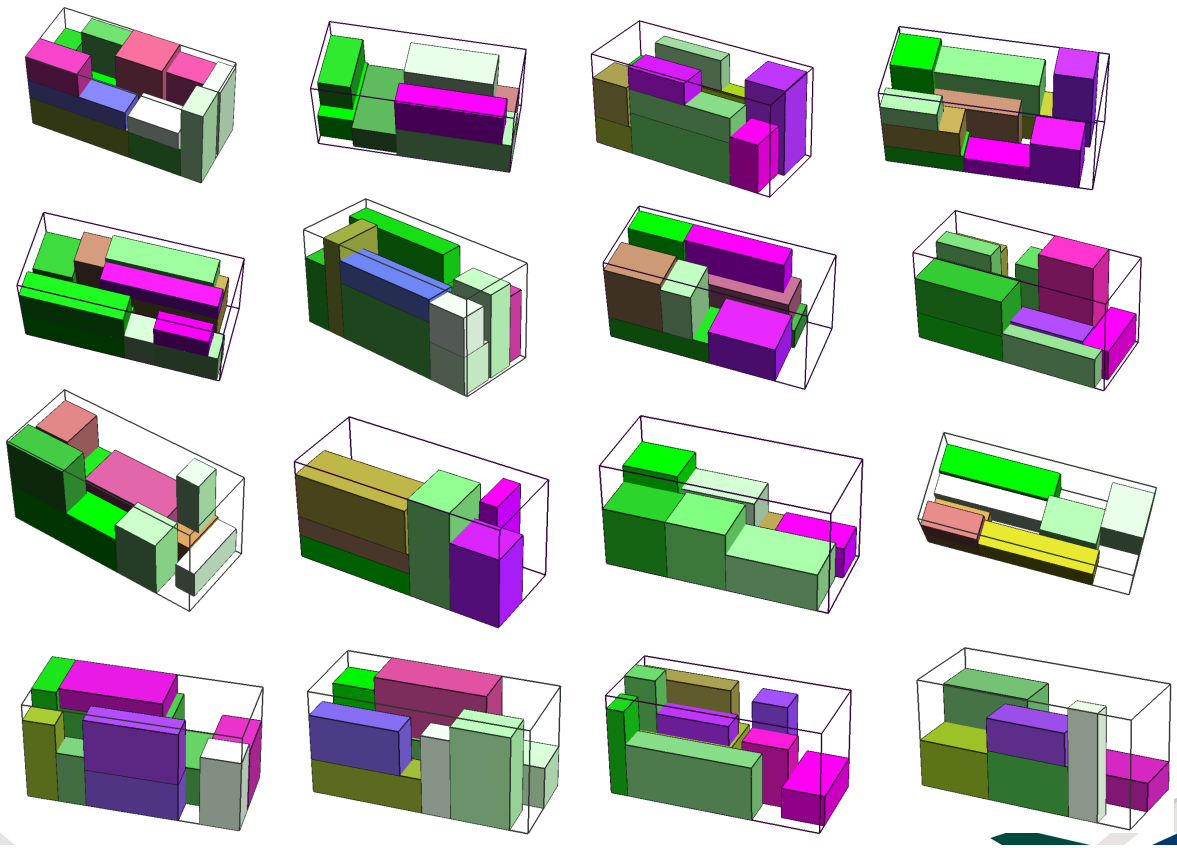


Figura 4.8: Patrones de empaquetamiento para cada una de las rutas del BKS encontrado para la instance 23.

Capítulo 5

Algoritmos Metaheurísticos para el Problema del Agente Viajero Considerando Polución

5.1. Introducción

La contaminación se está expandiendo de manera creciente en nuestras ciudades, y una de sus principales causas son las emisiones provenientes de los vehículos que transitan por las carreteras. Por lo tanto, reducir las emisiones de carbono es una meta fundamental que debe buscar alcanzarse en los problemas de enrutamiento de vehículos ([Bektaş and Laporte \(2011\)](#), [Lin et al. \(2014\)](#)). En [Bektaş and Laporte \(2011\)](#), fue introducido el Problema de Enrutamiento Considerando Polución PRP (por sus siglas en inglés *Pollution Routing Problem*); y se trata de una variante del VRP, en la cual la meta no es solamente minimizar la distancia de los viajes, si no también, minimizar la emisión de gases de efecto invernadero, consumo de combustible, tiempos de viaje y los costos asociados a los mismos. Se propone en este capítulo, un modelo de Programación Lineal Entero Mixto MILP (por sus siglas en inglés *Mixed Integer Linear Programming*), junto con un análisis de la forma como se afectan las diferentes medidas de rendimiento del enrutamiento de vehículos de esta variante, como

son: la distancia, la carga, las emisiones de gas y los costos de operación. En [Demir et al. \(2012\)](#) el modelo MILP fue extendido para permitir viajes de baja velocidad, y se propone una efectiva heurística de búsqueda adaptativa de vecindario amplia para el PRP. Un enfoque mateheurístico combina una búsqueda local, junto con el MILP para el PRP; obteniendo un mejor rendimiento que los algoritmos anteriormente propuestos.

Motivados por los estudios recientes de PRP, en [Cacchiani et al. \(2018\)](#) se propone el Problema del Agente Viajero Considerando Polución PTSP (por sus siglas en inglés *Pollution Traveling Salesman Problem*), en el cual se busca obtener un Tour Hamiltoniano que minimice una función de consumo de combustible (dependiente de la velocidad del vehículo y la carga que transporta), y los costos asociados al conductor. De manera más puntual, el PTSP corresponde a la versión de una sola ruta o tour del PRP como se modela en [Demir et al. \(2012\)](#). El PTSP es descrito formalmente en la Sección 5.2, donde también se presenta el modelo MILP correspondiente, mejorado con restricciones de eliminación de subtour explícitas. En [Cacchiani et al. \(2018\)](#), se propone un Algoritmo de Búsqueda Local Iterativa ILS (por sus siglas en inglés *Iterated Local Search Algorithm*) ([Lourenço et al. \(2010\)](#)), y se reportaron experimentos computacionales en instancias hasta con 50 clientes. El ILS es presentado en la Sección 5.3. La principal contribución de este trabajo son dos algoritmos mateheurísticos: (i) Un Algoritmo Genético Multi-operador MGA (por sus siglas en inglés *Multi-operator Genetic Algorithm*) y (ii) Un Algoritmo Híbrido HA (por sus siglas en inglés *Hybrid Algorithm*), basado en la combinación del ILS y el MGA, presentados en la Sección 5.4. Los algoritmos propuestos tienen la capacidad de encontrar soluciones de buena calidad en tiempos computacionales muy cortos. En la Sección 5.5, se reportan los experimentos computacionales realizados sobre las instancias de hasta 200 clientes, propuestas en [Demir et al. \(2012\)](#) y adaptadas en este estudio para el caso de un solo vehículo: en particular, acá se comparan los resultados obtenidos por el ILS, el MGA y el HA con aquellos encontrados por el algoritmo de *Cut-and-Branch*, basado en el modelo MILP mejorado presentado en la Sección 5.2. Finalmente, se plantean algunas conclusiones en la Sección 5.6.

5.2. Descripción y Formulación del Problema

El PTSP es definido como un grafo completo y dirigido $G = (\mathcal{N}, \mathcal{A})$, donde $\mathcal{N} = \{0, \dots, n\}$ es el conjunto de nodos, 0 es el depósito y \mathcal{A} es el conjunto de arcos. La distancia del nodo i al nodo j , se denota como d_{ij} . El conjunto $\mathcal{N}_0 = \mathcal{N} \setminus \{0\}$ es el conjunto de clientes. Cada cliente $i \in \mathcal{N}_0$ tiene una demanda no negativa q_i , y un tiempo de servicio t_i . Se define en este trabajo $D = \sum_{i \in \mathcal{N}_0} q_i$ como la capacidad del vehículo, y u_d el salario del conductor por unidad de tiempo. Considerando una función de velocidad discretizada definida por $|R|$ con niveles de velocidad no decrecientes \bar{v}^r ($r \in \mathcal{R}$), donde cada $r \in \mathcal{R}$, corresponde a un intervalo de velocidad en el rango $[v^l, v^u]$ (donde v^l y v^u corresponden a los límites inferior y superior de velocidad de viaje del vehículo).

Se adopta la expresión de consumo de combustible propuesta en [Demir et al. \(2012\)](#), la cual extiende la expresión que se presenta en [Bektaş and Laporte \(2011\)](#), permitiendo velocidades inferiores a 40 *km/hora*. Para un mayor detalle, se recomienda al lector explorar estos documentos para conocer la forma como esta expresión es derivada. Para un arco dado $(i, j) \in \mathcal{A}$ de longitud d_{ij} , atravesado a una velocidad v por un vehículo transportando una carga $M = w + f_{ij}$, donde w es el peso del vehículo vacío (peso neto) y f_{ij} es la carga transportada por el vehículo en este arco; el consumo de combustible entonces puede ser expresado como:

$$F(v) = \lambda k N V d_{ij} / v + \lambda \beta \gamma d_{ij} + \lambda w \gamma \alpha_{ij} d_{ij} + \lambda \gamma \alpha_{ij} f_{ij} d_{ij} \quad (5.1)$$

Donde $\lambda = \xi / \kappa \psi$ y $\gamma = 1 / 1000 \eta_{tf} \eta$ son constantes, $\beta = 0,5 C_d \rho A$ es una constante específica del vehículo, $\alpha_{ij} = \tau + g \sin \theta_{ij} + g C_r \cos \theta_{ij}$ es una constante específica del arco dependiendo del ángulo de inclinación o pendiente de la carretera θ_{ij} del arco (i, j) , y la aceleración τ (m/s^2). Todos los parámetros y valores han sido tomados de [Demir et al. \(2012\)](#), y se presentan en la Tabla 5.1. En particular, los primeros dos términos de (5.1) representan los requerimientos de energía que surgen con la velocidad inducida, mientras que los dos últimos términos representan los requerimientos energéticos inducidos por la carga transportada.

Tabla 5.1: Parámetros utilizados en el modelo PTSP.

Notación	Descripción Típica	Valores
w	Peso del vehículo vacío (kilogramos)	6350
ξ	Relación combustible-aire por unidad de masa	1
k	Factor de fricción del motor (kilojoule/rev/litro)	0.2
N	Velocidad del motor (rev/segundo)	33
V	Desplazamiento del motor (litros)	5
g	Constante gravitacional (metros/segundo ²)	9.81
C_d	Coefficiente de arrastre aerodinámico	0.7
ρ	Densidad del aire (kilogramo/metro ³)	1.2041
A	Área de superficie frontal (metro ²)	3.912
C_r	Coefficiente de resistencia a la rodadura	0.01
η_{tf}	Eficiencia de la caja de cambios del vehículo	0.4
η	Parámetro de eficiencia de los motores diesel	0.9
f_d	Salario de conductor por unidad de tiempo (£/segundo)	0.0022
κ	Valor típico de calor para combustible diesel (kilojoule/gramo)	44
ψ	Factor de conversión (gramo/segundo a litro/segundo)	737
v^l	Límite inferior de velocidad (metro/segundo)	5.5 (o 20 kilómetros/hora)
v^u	Límite superior de velocidad (metro/segundo)	25 (o 90 kilómetros/hora)

El PTSP busca determinar el Tour Hamiltoniano de mínimo costo que inicia desde el depósito y visita cada cliente exactamente una vez atendiendo su demanda, donde el costo está dado por la suma del consumo de combustible y el salario del conductor. Se introducen en este trabajo las siguientes variables de decisión: (i) variables binarias x_{ij} asumiendo valor 1 si el arco $(i, j) \in \mathcal{A}$ es recorrido; las variables no negativas f_{ij} representan la cantidad de flujo (es decir, la carga presente en el vehículo) viajando por el arco $(i, j) \in \mathcal{A}$; (iii) variables binarias z_{ij}^r asumiendo el valor 1 si el arco $(i, j) \in \mathcal{A}$ es atravesado a la velocidad $r \in \mathcal{R}$. El modelo MILP para el PTSP se presenta a continuación:

$$\text{Min} \quad \sum_{(i,j) \in \mathcal{A}} \lambda k N V d_{ij} \sum_{r \in \mathcal{R}} z_{ij}^r / \bar{v}^r + \sum_{(i,j) \in \mathcal{A}} \lambda \beta \gamma d_{ij} \sum_{r \in \mathcal{R}} z_{ij}^r (\bar{v}^r)^2 \quad (5.2)$$

$$+ \sum_{(i,j) \in \mathcal{A}} \lambda w \gamma \alpha_{ij} d_{ij} x_{ij} + \sum_{(i,j) \in \mathcal{A}} \lambda \gamma \alpha_{ij} d_{ij} f_{ij} \quad (5.3)$$

$$+ f_d \left(\sum_{(i,j) \in \mathcal{A}} \sum_{r \in \mathcal{R}} (d_{ij} / \bar{v}^r) z_{ij}^r + \sum_{i \in \mathcal{N}_0} t_i \right) \quad (5.4)$$

sujeto a

$$\sum_{j \in \mathcal{N}_0} f_{0j} = D \quad (5.5)$$

$$\sum_{j \in \mathcal{N}_0} f_{j0} = 0 \quad (5.6)$$

$$\sum_{j \in \mathcal{N}} x_{ij} = 1, \quad \forall i \in \mathcal{N} \quad (5.7)$$

$$\sum_{i \in \mathcal{N}} x_{ij} = 1, \quad \forall j \in \mathcal{N} \quad (5.8)$$

$$\sum_{j \in \mathcal{N}} f_{ji} - \sum_{j \in \mathcal{N}} f_{ij} = q_i, \quad \forall i \in \mathcal{N}_0 \quad (5.9)$$

$$q_j x_{ij} \leq f_{ij} \leq (D - q_i) x_{ij}, \quad \forall (i, j) \in \mathcal{A} \quad (5.10)$$

$$\sum_{r \in \mathcal{R}} z_{ij}^r = x_{ij}, \quad \forall (i, j) \in \mathcal{A} \quad (5.11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{A} \quad (5.12)$$

$$f_{ij} \geq 0, \quad \forall (i, j) \in \mathcal{A} \quad (5.13)$$

$$z_{ij}^r \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{A}, \forall r \in \mathcal{R} \quad (5.14)$$

La función objetivo consiste en tres componentes principales para minimizar: (5.2) y (5.3) representan el consumo de combustible, como se define en (5.1), teniendo en cuenta, respectivamente, la energía requerida por las variaciones de velocidad y la energía utilizada para llevar el vehículo vacío y la carga presente en el vehículo, mientras que (5.4) corresponde al salario del conductor, donde el término que está por fuera de los paréntesis corresponde a la duración total del tour, la cual depende de la velocidad a la que se viaja por los arcos y en los tiempos de servicio en cada uno de los clientes. Las restricciones (5.5) y (5.6) aseguran, respectivamente, que el vehículo sale completamente lleno y retorna vacío al depósito. Las restricciones (5.7) y (5.8) garantizan que cada nodo es visitado exactamente una vez. Las restricciones (5.9) y (5.10) definen la carga presente en el vehículo durante cada arco

visitado (e implícitamente prohíbe subtours). Finalmente, las restricciones de (5.11) enlazan las variables x y z imponiendo que un solo nivel de velocidad es seleccionado para cada arco utilizado $(i, j) \in \mathcal{A}$, y las restricciones (5.12)-(5.14) definen los dominios de las variables.

En este trabajo se adicionan al modelo presentado entre (5.2)-(5.14), la eliminación explícita de subtours SECs (por sus siglas en inglés de *Explicit Subtour Elimination Constraints*), como se propone en Dantzig et al. (1954) para el TSP Asimétrico:

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N} \setminus \mathcal{S}} x_{ij} \geq 1, \quad \mathcal{S} \subseteq \mathcal{N}, \quad \{0\} \in \mathcal{S}, \quad |\mathcal{S}| \geq 2. \quad (5.15)$$

La solución óptima de la relajación LP del modelo (5.2)-(5.15) es utilizada para obtener buenas soluciones factibles para los tres algoritmos metaheurísticos propuestos. Adicionalmente, el modelo (5.2)-(5.15) es utilizado como punto de referencia en los experimentos computacionales, para evaluar el comportamiento de los algoritmos propuestos. Puntualmente, se resuelve en este estudio el modelo (5.2)-(5.15) con un algoritmo *Cut-and-Branch*, en el cual las SECs son separadas, en el nodo raíz, utilizando el procedimiento de separación propuesto en Padberg and Rinaldi (1990), y el *solver* CPLEX es utilizado para obtener soluciones enteras del MILP.

Es importante anotar, que los algoritmos metaheurísticos determinan solamente la secuencia de clientes de una solución factible del PTSP, es decir, los valores de las variables correspondientes x_{ij} del modelo (5.2)-(5.15). El valor de la función objetivo asociada (5.2)-(5.4) puede ser fácilmente computado definiendo, para cada arco utilizado (i, j) , los valores correspondientes de la carga f_{ij} y el intervalo de velocidad r que produce el mínimo consumo de combustible.

5.3. Algoritmo de Búsqueda Local Iterativa

El pseudocódigo del ILS se presenta en el Algoritmo 6. El primer paso (líneas 1–8) consiste en resolver iterativamente el modelo de Programación Lineal (LP, por sus siglas en inglés

Linear Programming), resultante de la relajación del modelo (5.2)-(5.15); aplicando el proceso de separación propuesto en Padberg and Rinaldi (1990) para detectar el conjunto \mathcal{S} que contiene los subtours (es decir, un subconjunto de nodos \mathcal{S} para el cual el SEC (5.15)) correspondiente es violado): si existe uno, entonces el corte correspondiente SEC (5.15) es añadido al modelo y la relajación LP es resuelta nuevamente. Una vez la solución óptima x de la relajación LP ha sido obtenida, es utilizada en la línea (9) para construir un subtour como se describe a continuación. Inicialmente, se define el depósito 0 como el nodo inicial h . Luego, iterativamente, se selecciona un nodo j que aún no ha sido visitado, donde $x_{hj} + x_{jh}$ tiene el valor más alto: el arco (h, j) es añadido al tour, y j se convierte en el nuevo nodo inicial h . Si, para todos los nodos sin visitar i , se tiene $x_{hi} + x_{ih} = 0$, luego se selecciona el nodo sin visitar j con el valor más pequeño de la distancia original d_{hj} . El procedimiento se repite hasta que se obtiene un tour completo y factible. Se le denomina entonces s^* la solución óptima localmente y s^{**} la mejor solución actual. El siguiente ciclo (líneas 10–33) contienen tres fases: *perturbación*, *búsqueda local* y *criterio de aceptación*.

Para perturbar la mejor solución actual s^{**} (líneas 11–16) se aplica, con una probabilidad del 80 % un movimiento de doble-puente, o un movimiento de mezcla de subtours en caso contrario. El movimiento de doble-puente consiste en remover aleatoriamente cuatro aristas que presenten nodos disjuntos (A,B), (C,D), (E,F), (G,H), y reconectarlos como (A,F), (G,D), (E,B), (C,H). El movimiento de mezcla de subtours selecciona aleatoriamente un camino del tour y cambiar aleatoriamente el orden de sus nodos. Después de realizar la perturbación, se obtiene la solución s' .

La siguiente fase (líneas 17–27) es un procedimiento de búsqueda local que es aplicado a la solución óptima s^* ; con probabilidad del 80 % se aplica el movimiento 2-opt, de lo contrario, se realiza una intercambio de mejora. El primer movimiento consiste entonces en la ejecución del procedimiento 2-opt utilizando como costo de cada arco $(i, j) \in \mathcal{A}$ la distancia original d_{ij} y deteniendo el procedimiento cuando sucede la primera mejora. El segundo movimiento requiere del intercambio de dos nodos seleccionados aleatoriamente en el tour: si se obtiene una mejora del valor de la función objetivo PTSP (5.2)-(5.4), entonces se realiza el intercambio, de lo contrario, se mantiene el tour original. Este procedimiento es

ejecutado $|\mathcal{N}|$ veces. Después de aplicar el procedimiento de búsqueda local, se obtiene una nueva solución s'' , considerando la función ϕ que genera el valor de la función objetivo PTSP (5.2)-(5.4).

La última etapa (líneas 28–30) consiste en el criterio de aceptación (revisión del histórico): si s^{**} no ha sido mejorado durante 10 iteraciones, entonces se aplica un paso adicional de búsqueda local sobre s^{**} ejecutando el procedimiento 2-opt. Este utiliza como costos de los arcos las distancias originales $d_{ij}(i, j) \in \mathcal{A}$, pero cada vez que una mejora es posible, se revisa si el valor de la función objetivo PTSP (5.2)-(5.4) es también mejorado, y se acepta el cambio solamente en este caso.

Finalmente, la mejor solución entre s^* y s^{**} es almacenada en s^{**} . La condición de terminación es alcanzada cuando I iteraciones son realizadas ($I = 5000$ en los experimentos computacionales realizados en este estudio).

Algorithm 6 Iterated Local Search

Data: Instance Parameters**Result:** Feasible Tour

```
1  $\mathcal{F} \leftarrow$  empty cut family
2 repeat
3    $x \leftarrow$  solution of the LP-relaxation of (5.2)-(5.15) with respect to the cut family  $\mathcal{F}$ 
4    $\mathcal{S} \leftarrow$  violated cut found by the separation-procedure( $x$ )
5   if  $\mathcal{S} \neq \emptyset$  then
6     | add cut  $\mathcal{S}$  to the family  $\mathcal{F}$ 
7   end
8 until  $\mathcal{S} = \emptyset$ ;
9  $s^*, s^{**} \leftarrow$  build-feasible-tour( $x$ )
10 repeat
11    $r' \leftarrow \text{rnd}(0, 1)$ 
12   if  $r' < 0,8$  then
13     |  $s' \leftarrow$  double-bridge-move( $s^{**}$ )
14   else
15     |  $s' \leftarrow$  scramble-subtour( $s^{**}$ )
16   end
17    $r'' \leftarrow \text{rnd}(0, 1)$ 
18   if  $r'' < 0,8$  then
19     |  $s'' \leftarrow$  2-opt-move( $s^*$ )
20   else
21     |  $s'' \leftarrow$  exchange-improvement( $s^*$ )
22   end
23   if  $\phi(s') < \phi(s'')$  then
24     |  $s^* \leftarrow s'$ 
25   else
26     |  $s^* \leftarrow s''$ 
27   end
28   if check-history( $\phi(s^{**})$ ) then
29     |  $s^* \leftarrow$  2-opt-improvement( $s^{**}$ )
30   end
31   if  $\phi(s^*) < \phi(s^{**})$  then
32     |  $s^{**} \leftarrow s^*$ 
33   end
33 until termination condition;
```

5.4. Algoritmo Genético Multi-operador y Algoritmo Híbrido

Los Algoritmos Genéticos son metaheurísticas efectivas, y han sido exitosamente aplicadas para resolver el ATSP y en muchas de sus variantes ([Potvin \(1996\)](#), [Larranaga et al. \(1999\)](#), [Moon et al. \(2002\)](#), [Snyder and Daskin \(2006\)](#), [Yuan et al. \(2013\)](#), [Groba et al. \(2015\)](#)). Generalmente, estos algoritmos utilizan solamente operadores individuales de cruzamiento y mutación, sin tener en cuenta el potencial que presenta la sinergia de los multi-operadores. Sin embargo, los operadores de cruzamiento y mutación se pueden complementar mutuamente, generando la mencionada sinergia que brinda mejores resultados que los que se alcanzan con operadores individuales. ([Li et al. \(2014\)](#), [Contreras-Bolton and Parada \(2015\)](#)). Aprovechando las ventajas de los multi-operadores, se propone un enfoque basado en un MGA.

En la primera parte de esta sección, se describe el MGA (ver Algoritmo [7](#)) y sus componentes: representación, generación de la población inicial, la función de evaluación, los operadores de cruzamiento y mutación y los parámetros genéticos presentados en las subsecciones [5.4.1](#), [5.4.2](#), [5.4.3](#), [5.4.4](#) y [5.4.5](#). El Algoritmo Híbrido (HA, por sus siglas en inglés *Hybrid Algorithm*), resultante de la combinación del MGA con el ILS, se presenta en la subsección [5.4](#).

Algorithm 7 Multi-operator Genetic Algorithm

Data: Instance Parameters**Result:** Best Individual

```
1  $\mathcal{F} \leftarrow$  empty cut family;  $l = 1$ 
2 repeat
3    $x \leftarrow$  solution of the LP-relaxation of (5.2)-(5.15) with respect to the cut family  $\mathcal{F}$ 
4   if  $l = k^1$  then
5      $x^1 \leftarrow x$ 
6   else if  $l = k^2$  then
7      $x^2 \leftarrow x$ 
8   else if  $l = k^3$  then
9      $x^3 \leftarrow x$ 
10  end
11   $S \leftarrow$  violated cut found by the separation-procedure( $x$ )
12  if  $S \neq \emptyset$  then
13    add cut  $S$  to the family  $\mathcal{F}$ 
14  end
15 until  $S = \emptyset$  or  $l = k^3$ ;
16 for  $t \leftarrow 1$  to numbers of individuals do
17    $r \leftarrow \text{rnd}(0, 1)$ 
18   if  $r < 0,2$  then
19      $I_{0,t} \leftarrow \text{random-heuristic}()$ 
20   else if  $0,2 \leq r < 0,4$  then
21      $I_{0,t} \leftarrow \text{nearest-neighbor-heuristic}()$ 
22   else if  $0,4 \leq r < 0,6$  then
23      $I_{0,t} \leftarrow \text{LP-based-heuristic}(x^1)$ 
24   else if  $0,6 \leq r < 0,8$  then
25      $I_{0,t} \leftarrow \text{LP-based-heuristic}(x^2)$ 
26   else if  $0,8 \leq r$  then
27      $I_{0,t} \leftarrow \text{LP-based-heuristic}(x^3)$ 
28   end
29    $I_{0,t} \leftarrow \text{2-opt-improvement}(I_{0,t})$ 
30   evaluation of individual  $I_{0,t}$ 
31 end
32 for  $g \leftarrow 1$  to maximum number of generations do
33   for  $t \leftarrow 1$  to numbers of individuals - 1 do
34      $(j, k) \leftarrow \text{selection}()$ 
35      $(I_{g,t}, I_{g,t+1}) \leftarrow \text{crossover}(I_{g-1,j}, I_{g-1,k}, \text{OX2 or DPX or HX or UNN})$ 
36      $I_{g,t} \leftarrow \text{mutation}(I_{g,t}, \text{EM or GSTM or 2opt or 3opt})$ 
37      $I_{g,t+1} \leftarrow \text{mutation}(I_{g,t+1}, \text{EM or GSTM or 2opt or 3opt})$ 
38     evaluation of individual  $I_{g,t}$  and  $I_{g,t+1}$ 
39   end
40   elitism( $I_g, I_{g-1}$ )
41 end
```

En la etapa inicial (líneas 1–15), la solución de la relajación LP del modelo (5.2)-(5.15) es obtenida mediante un procedimiento similar al descrito en las líneas 1–8 del Algoritmo 6, con la diferencia de que como máximo se ejecutarán k^3 iteraciones del ciclo haga-mientras. Adicionalmente, después de k^1 , k^2 y k^3 iteraciones, la solución x del LP, encontrada en la línea 3 es almacenada, respectivamente, en las soluciones x^1 , x^2 y x^3 . Los valores de k^1 , k^2 y k^3 son parámetros dados. Luego, una población de individuos es generada (aplicando cinco procedimientos basados en tres métodos diferentes) y evaluados utilizando las instrucciones reportadas en las líneas 16–31 (nótese que cada procedimiento de inicialización tiene la misma probabilidad de ocurrir). Posteriormente, el ciclo principal del algoritmo, presentado en las líneas 32–41, es responsable de generar una nueva población a partir de la actual, a través de múltiples operadores de cruzamiento y mutación, como se describe en las líneas 34–37. La selección es realizada en un torneo de tres individuos (Eiben and Smith (2015)). También se aplica elitismo, donde los mejores padres de la anterior población, reemplazan el 10 % de los peores individuos generados en la población actual (línea 40). El individuo que presenta el valor mínimo de la función objetivo (5.2)-(5.4) es almacenado como el «mejor individuo».

5.4.1. Representación y función *fitness*

Una representación tipo permutación es utilizada, donde cada individuo corresponde a un tour Hamiltoniano. La función objetivo (5.2)-(5.4) es utilizada como función *fitness*, pero se computa la función objetivo para cada nivel de velocidad $r \in \mathcal{R}$ tomando como valor *fitness*, el valor más pequeño de los valores que fueron computados.

5.4.2. Población inicial

La población inicial es calculada utilizando cinco procedimientos basados en tres métodos heurísticos que serán descritos a continuación. Cada procedimiento tiene la misma probabilidad de ocurrencia. Luego se aplica el procedimiento 2-opt (descrito en la Sección 5.3) para generar individuos e intentar mejorarlos. Las heurísticas consideradas son las siguientes:

- Heurística-Aleatoria: El tour es generado aleatoriamente.
- Heurística-Vecino-Más-Cercano: El tour es generado aplicando la heurística del vecino más cercano (NNH, por sus siglas en inglés *Nearest Neighbor Heuristic*) (Flood (1956)), que inicia aleatoriamente de uno de los nodos presentes en \mathcal{N} .
- Heurística-LP: La heurística, descrita en el Algoritmo 8, genera una solución factible utilizando un NNH aleatorizado, basado en la solución del modelo LP relajado (x_{ij}), correspondiendo a una de las tres soluciones x^1 , x^2 y x^3 obtenidas en el paso inicial (líneas 1–15) del Algoritmo 7. La aleatorización ocurre en la línea 7.

Algorithm 8 LP-based-heuristic

Data: Instance Parameters

Result: Complete tour T

```

1  $h = 0$ 
2  $T \leftarrow \{h\}$ 
3 repeat
4    $max \leftarrow 0$ 
5   for  $k \in \mathcal{N} \setminus T$  do
6      $r \leftarrow rnd(0, 1)$ 
7     if  $x_{hk} + x_{kh} > max$  and  $r < 0,5$  then
8        $max \leftarrow x_{hk} + x_{kh}$ 
9        $j \leftarrow k$ 
10    end
11  end
12  if  $max = 0$  then
13     $j \leftarrow$  node corresponding to the smallest  $d_{hj}$  (with  $j \in \mathcal{N} \setminus T$ ).
14  end
15   $T \leftarrow T \cup \{j\}$ 
16   $h \leftarrow j$ 
17 until a complete tour  $T$  is obtained;
```

5.4.3. Operadores de cruzamiento

Cuatro operadores de cruzamiento son utilizados: *Order Based Crossover* (OX2) (Syswerda (1991)), *Distance Preserving Crossover* (DPX) (Reisleben et al. (1996)), *Heuristic Crossover* (HX) (Grefenstette et al. (1985)) y *Uniform Nearest Neighbor* (UNN) (Buriol et al. (2004)), con probabilidades de 10 %, 45 %, 20 % y 25 % respectivamente.

- OX2: Selecciona aleatoriamente un subconjunto de posiciones consecutivas del primer padre, de acuerdo al orden de los nodos en las posiciones seleccionadas de este padre, impone dicho conjunto de nodos en el segundo padre.
- DPX: Los nodos contenidos en el primer padre, son copiados en la descendencia, y todos los arcos que no son comunes con el segundo padre se eliminan, generando a un conjunto de caminos desconectados (nótese que algunos de los caminos pueden consistir de un solo nodo). Estos caminos son reconectados sin utilizar los arcos contenidos solamente en uno de los padres. Puntualmente, dado un camino que termina en el nodo i , el nodo vecino disponible más cercano k entre los nodos iniciales de los caminos restantes, es tomado y el arco (i, k) es añadido al tour, a menos que (i, k) está contenido en uno de los dos padres. El procedimiento es repetido hasta que todos los caminos han sido reconectados en un tour.
- HX: inicialmente, un nodo es seleccionado aleatoriamente para establecerse como el nodo actual de la descendencia. Luego, se selecciona la arista (sin dirección) más barata (entre un grupo no mayor a 4) que conecta el nodo seleccionado, con un nodo que no haya sido visitado y que pertenezca al tour de los padres. Si ninguna de las aristas parentales lleva a un nodo sin visitar, se selecciona una arista de manera aleatoria. Esto se repite hasta que se construya un tour completo.
- UNN: Al iniciar, todos los arcos en común entre ambos padres son copiados dentro de la descendencia, obteniendo un conjunto de caminos desconectados. Los arcos restantes son insertados de la siguiente manera: para cada nodo i que sea nodo final de un camino, un valor verdadero o falso es generado aleatoriamente con la misma probabilidad de

ocurrencia. Si el valor generado es verdadero (respectivamente en caso de que el valor sea falso), luego, si no se han violado restricciones, el arco que enlaza el nodo i con el siguiente nodo en el padre A (respectivamente con el padre B), es copiado en la descendencia. Si una violación ocurre en cualquiera de los dos casos, entonces el arco del otro padre es considerado. Los fragmentos de tour resultantes son reparados utilizando el algoritmo NNH.

5.4.4. Operadores de mutación

Cuatro operadores de mutación fueron utilizados: *exchange mutation* (EM), *Greedy Sub Tour Mutation* (GSTM), 2-opt y 3-opt, con probabilidades del 10 %, 35 %, 20 % y 35 %, respectivamente. EM está basada en el operador propuesto en [Banzhaf \(1990\)](#). GSTM combina operadores clásicos (como «mutación de inversión simple» y «mutación por mezcla») y técnicas voraces utilizando diferentes parámetros. La estructura paramétrica del operador previene atascos en óptimos locales, y alcanza soluciones de buena calidad en cortos tiempos computacionales utilizando métodos de búsqueda golosos, realizando perturbaciones aleatorias en esas soluciones (de la misma forma que los operadores de mutación clásicos), y luego aplicándoles nuevamente un método voraz, se detalla en ([Albayrak and Allahverdi \(2011\)](#)). El operador 3-opt se implementó considerando todos los pares de aristas, y seleccionando, por cada par, una tercera arista de manera aleatoria en solamente diez intentos, en lugar de considerar todas las aristas faltantes.

5.4.5. Parámetros del algoritmo genético

Los parámetros involucrados en el MGA son: el tamaño de la población, máximo número de ejecuciones generacionales, y las probabilidades de mutación y cruzamiento. La definición adecuada de los parámetros está directamente relacionada con el rendimiento computacional de los algoritmos evolutivos. Basado en la literatura existente ([Eiben et al. \(1999\)](#), [Eiben et al. \(2007\)](#)), se establecen los siguientes parámetros: el tamaño de la población en 150 individuos, la probabilidad de cruzamiento en 0,9, y la probabilidad de mutación en 0,2.

Las probabilidades de los operadores de cruzamiento y mutación se basaron en experimentos computacionales preliminares.

5.4.6. Algoritmo Híbrido

El tercer algoritmo mateheurístico que se propone en este trabajo es el Algoritmo Híbrido HA, obtenido por la combinación entre el MGA y el ILS. En este, se aplica el MGA (descrito en el Algoritmo 7), y define s^* y s^{**} como la mejor solución encontrada. Luego se aplican tres fases de mejora del ILS: *perturbación*, *búsqueda local* y *criterio de aceptación* (como se especifica en las líneas 10–33 del Algoritmo 6).

5.5. Experimentos Computacionales

En este trabajo se utilizaron conjuntos de instancias de prueba con 10, 15, 20, 25, 50, 75, 100, 150 y 200 clientes, propuestas en [Demir et al. \(2012\)](#) para el PRP, y se adaptaron al PTSP. Concretamente, para hacer las instancias factibles para un sólo vehículo, se eliminaron las restricciones correspondientes a las ventanas de tiempo para cada cliente, y se emplea un vehículo con una capacidad $D = \sum_{i \in \mathcal{N}_0} q_i$. Adicionalmente, se consideran las instancias generadas en [Cacchiani et al. \(2018\)](#), con 30, 35, 40 y 45 clientes, obtenidas de las instancias de prueba con 50 clientes, considerando, respectivamente, los primeros 30, 35, 40 y 45 clientes. Cada conjunto de instancias contiene 20 instancias. Los algoritmos fueron implementados en C++, y todos los experimentos fueron ejecutados en un *Intel Core i7-6900K* con 16-Núcleos de 3.20GHz y 66 GB RAM (mono hilo). Se utilizó CPLEX 12.7.1 como *solver* para el LP y el MILP, y se estableció un tiempo límite, para el algoritmo *Cut-and-Branch*; dos horas para instancias hasta 50 clientes, y 5 horas para las instancias más grandes. Se fijó $k^1 = 10$, $k^2 = 30$ y $k^3 = 50$ para instancias hasta 100 clientes, y $k^1 = 20$, $k^2 = 60$ y $k^3 = 100$ para las demás instancias.

Los resultados obtenidos por el algoritmo *Cut-and-Branch* (C&B) y por el ILS son reportados en la Tabla 5.2, mientras que los resultados obtenidos por el MGA y el HA se muestran en

la Tabla 5.3. También se resolvió el modelo (5.2)-(5.14) aplicando directamente el *solver* CPLEX al MILP, pero el algoritmo C&B consigue encontrar un número mayor de soluciones óptimas y tiene gaps y tiempos computacionales promedio menores. Por lo tanto, se reportan solamente los resultados obtenidos por el algoritmo C&B.

Cada fila en las Tablas 5.2 y 5.3 corresponden a un conjunto de instancias y muestran resultados promedio sobre las 20 instancias en el conjunto. En ambas tablas, para cada fila, primero se reporta el número de clientes ($\#Cust$) en el conjunto correspondiente, y el valor promedio ($Best$) de las mejores soluciones conocidas, es decir, el promedio de los valores de la mejor solución encontrada por cualquiera de los métodos considerados (C&B, ILS, MGA y HA) en cada una de las 20 instancias en el conjunto. En el caso del algoritmo C&B, se reporta, para cada fila, el promedio de los valores de las soluciones enteras finales (UB), el promedio de los gaps porcentuales ($GapLB\%$) entre el valor de la solución entera y el *lower-bound* correspondiente obtenido, para cada instancia del conjunto, al final del proceso de solución C&B, el número de mejores soluciones ($\#B$) encontradas, el promedio de los porcentajes de Gap ($Gap\%$) entre los valores de las soluciones enteras finales y las mejores soluciones conocidas correspondientes, y el tiempo de cómputo promedio ($Tiempo$) expresado en segundos. Para el ILS, el MGA y el HA, se realizaron diez corridas para cada instancia en cada uno de los conjuntos, y se reportan el *promedio* y el *mínimo* resultado obtenido sobre las 10 corridas realizadas. Se especifica el promedio de los valores de las soluciones (Val), el promedio de los gaps porcentuales ($Gap\%$) con respecto a los valores de las mejores soluciones conocidas, el número de las mejores soluciones ($\#B$) encontradas, y los tiempos computacionales ($Time$) en segundos. Nótese que el tiempo computacional obtenido, para cada instancia, el «mínimo» resultado sobre las 10 corridas, es diez veces el valor reportado en la columna de tiempo para el resultado «promedio» correspondiente. Finalmente, la última fila reporta, para cada columna, el valor promedio sobre todos los conjuntos de instancias. Nótese también que en algunos casos, incluso cuando el valor Val es el mismo que el promedio del valor de la mejor solución conocida *Mejor*, el número de mejores soluciones encontradas es menor que 20, dado que se reportan los resultados promedio con dos dígitos decimales.

Se observa que el algoritmo C&B es capaz de probar la optimalidad de la solución encontrada

por todas las instancias hasta 25 clientes, como se indica en los valores $GapLB\%$ iguales a 0.00, en tiempos computacionales cortos (en promedio como máximo 22 segundos de cómputo). Para instancias de mayor tamaño, el gap porcentual con respecto al límite inferior se incrementa significativamente, alcanzando en promedio 15.40 % para las instancias con 200 clientes. Adicionalmente, los tiempos computacionales también se incrementan y el tiempo límite es alcanzado por todas las instancias con al menos 45 clientes. Observando las columnas $\#B$ y $Gap\%$, se aprecia que la mejor solución es encontrada por el algoritmo C&B para todas las instancias hasta 35 clientes, y para la mayoría de las instancias hasta 50 clientes. Sin embargo, en instancias de mayor tamaño, sólo unas pocas BKS son obtenidas por el algoritmo C&B, y no se encuentra ninguna solución de buena calidad para instancias con 200 clientes.

El algoritmo ILS obtiene buenos resultados particularmente cuando se consideran los valores mínimos de las soluciones alcanzadas a través de las 10 corridas realizadas para cada instancia: de hecho, obtiene todos excepto una de las mejores soluciones conocidas para instancias hasta 50 clientes. Para instancias con más de 100 clientes, incluso considerando el valor de la solución mínima sobre las diez corridas, se observa que sólo unas pocas BKS son encontradas (4 como máximo). De los resultados obtenidos por el ILS en los promedios sobre las 20 corridas, se observa que el número de mejores soluciones conocidas encontrados es cercano a solamente 20 instancias hasta 20 clientes. Sin embargo, también se aprecia que los gaps porcentuales, tanto para el *promedio* y los resultados *mínimos*, son pequeños (a lo sumo 1.25 %), mostrando que, aunque la mejor solución conocida no es encontrada, el resultado arrojado presenta buena calidad. Adicionalmente, se aprecia que los tiempos computacionales son muy cortos, arrojando en promedio 13.83 segundos.

Los valores mínimos de las soluciones alcanzadas a través de las 10 corridas, el rendimiento del MGA es similar al del ILS. No obstante, se aprecia en los resultados *promedio* alcanzados sobre las 10 ejecuciones, que el MGA obtiene una mayor cantidad de BKS con respecto al ILS. Además, el gap porcentual (a lo sumo 0.93 % y 0.23 % en promedio) es más pequeño que el que presenta el ILS. El tiempo computacional del MGA es similar al del ILS, siendo 14.49 segundos en promedio.

Los mejores rendimientos son alcanzados por el HA. En efecto, HA es capaz de encontrar la mejor solución conocida para todas las instancias de hasta 75 clientes, cuando se consideran los resultados *mínimos* alcanzados en las 10 corridas. Además, para las instancias más grandes, se aprecia que el rendimiento del HA es claramente mejor que el de los otros algoritmos, dado que el HA obtiene las BKS para 19 instancias con 100 clientes, 14 instancias con 150 clientes, y 18 instancias con 200 clientes. El gap porcentual con respecto al valor de la mejor solución conocida es muy pequeño (0.01 % en promedio). Por lo tanto, este algoritmo mateheurístico es más estable que los otros presentados. Observando los resultados *promedio* alcanzados sobre las 10 ejecuciones, se aprecia que el número de BKS encontrados es el mayor entre todos los algoritmos mateheurísticos (11.8 en promedio) y el gap porcentual es el más pequeño (0.14 % en promedio). Finalmente, se aprecia un incremento leve en el tiempo computacional, que de todas formas sigue siendo muy corto (15.47 segundos en promedio) comparado al del algoritmo C&B. En resumen, se puede concluir que el HA combina exitosamente los algoritmos MGA e ILS, y tiene la capacidad de mejorar notablemente los resultados obtenidos por los algoritmos C&B e ILS.

		C&B					ILS						
							<i>Promedio</i>				<i>Mínimo</i>		
#Clientes	Mejor	UB	GapLB %	#B	Gap %	Tiempo	Val	Gap %	#B	Tiempo	Val	Gap %	#B
10	150.64	150.64	0.00	20	0.00	0.29	150.64	0.00	19	0.02	150.64	0.00	20
15	215.69	215.69	0.00	20	0.00	0.58	215.69	0.00	18	0.06	215.69	0.00	20
20	288.56	288.56	0.00	20	0.00	2.61	288.60	0.02	17	0.11	288.56	0.00	20
25	311.45	311.45	0.00	20	0.00	22.19	311.92	0.13	14	0.21	311.45	0.00	20
30	417.52	417.52	0.05	20	0.00	911.08	417.85	0.08	10	0.42	417.52	0.00	20
35	493.21	493.21	1.00	20	0.00	4226.34	493.76	0.11	10	0.58	493.31	0.02	19
40	563.01	563.13	2.67	19	0.02	6495.88	563.48	0.08	5	0.79	563.01	0.00	20
45	643.86	644.22	4.40	16	0.05	7200.00	645.47	0.24	5	0.99	644.05	0.03	19
50	719.77	720.66	5.55	17	0.12	7200.00	721.05	0.18	4	1.28	719.77	0.00	20
75	1266.29	1268.85	9.02	13	0.19	18000.00	1271.15	0.38	3	3.74	1266.68	0.03	16
100	1802.50	1808.41	11.06	5	0.32	18000.00	1816.43	0.78	0	10.11	1804.87	0.13	7
150	3148.28	3166.51	13.51	2	0.59	18000.00	3184.43	1.15	0	37.99	3161.98	0.44	4
200	4461.07	4538.99	15.40	0	1.75	18000.00	4516.84	1.25	0	123.50	4478.46	0.39	2
Promedio	1113.99	1122.14	4.82	14.8	0.23	7543.00	1122.87	0.34	8.1	13.83	1116.61	0.08	15.9

Tabla 5.2: Resultados obtenidos por el Algoritmo Cut-and-Branch y por el Algoritmo ILS

		MGA							HA						
		<i>Promedio</i>				<i>Mínimo</i>			<i>Promedio</i>				<i>Mínimo</i>		
#Clientes	Mejor	Val	Gap %	#B	Tiempo	Val	Gap %	#B	Val	Gap %	#B	Tiempo	Val	Gap %	#B
10	150.64	150.65	0.00	19	0.06	150.64	0.00	20	150.64	0.00	20	0.08	150.64	0.00	20
15	215.69	215.69	0.00	19	0.11	215.69	0.00	20	215.69	0.00	20	0.15	215.69	0.00	20
20	288.56	288.56	0.00	20	0.19	288.56	0.00	20	288.56	0.00	20	0.26	288.56	0.00	20
25	311.45	311.54	0.03	17	0.32	311.45	0.00	20	311.45	0.00	20	0.39	311.45	0.00	20
30	417.52	417.61	0.03	17	0.57	417.52	0.00	20	417.57	0.02	17	0.66	417.52	0.00	20
35	493.21	493.48	0.05	13	0.81	493.30	0.02	19	493.29	0.01	15	0.92	493.21	0.00	20
40	563.01	563.20	0.03	12	1.08	563.01	0.00	20	563.10	0.02	15	1.17	563.01	0.00	20
45	643.86	644.44	0.09	7	1.36	644.09	0.04	17	644.24	0.06	9	1.53	643.86	0.00	20
50	719.77	720.51	0.10	10	1.80	719.77	0.00	19	720.19	0.06	12	1.98	719.77	0.00	20
75	1266.29	1269.63	0.26	3	5.03	1267.58	0.10	16	1267.61	0.10	6	5.32	1266.29	0.00	20
100	1802.50	1811.92	0.52	0	11.06	1805.24	0.15	7	1808.70	0.34	0	11.49	1802.81	0.02	19
150	3148.28	3177.14	0.93	0	39.67	3154.96	0.21	3	3167.29	0.61	0	44.26	3149.76	0.05	14
200	4461.07	4501.52	0.91	0	126.38	4468.83	0.17	3	4489.70	0.64	0	132.94	4461.18	0.00	18
Promedio	1113.99	1120.45	0.23	10.5	14.49	1115.43	0.05	15.7	1118.31	0.14	11.8	15.47	1114.13	0.01	19.3

Tabla 5.3: Resultados obtenidos por el Algoritmo Genético Multi-operador (MGA) y por el Algoritmo Híbrido HA

5.6. Conclusiones

En este trabajo se proponen dos algoritmos mateheurísticos para resolver el PTSP: un Algoritmo Genético MGA y un Algoritmo Híbrido HA, el cual combina el MGA con un ILS propuesto en un trabajo previo. El MGA presenta cuatro operadores de cruzamiento y cuatro operadores de mutación, los cuales permiten la generación de soluciones de buena calidad en tiempos de cómputo cortos. El HA aprovecha la combinación del MGA con el ILS, el cual aplica de manera iterativa perturbación y procedimientos de búsqueda local para posiblemente mejorar la incumbente del MGA. Ambos, el MGA y el HA, utilizan efectivamente la solución LP (Linear Programming) del modelo MILP (por sus siglas en inglés de *Mixed Integer Linear Programming*) para generar la población inicial.

Se probaron los dos algoritmos mateheurísticos propuestos en instancias hasta de 200 clientes, adaptadas de las propuestas en [Demir et al. \(2012\)](#) para el Problema de Enrutamiento considerando Polución (PRP). Para evaluar sus rendimientos, se compararon los resultados obtenidos con los cálculos de los algoritmos ILS y C&B, en el cual, las restricciones de eliminación de subtour son agregadas en el nodo raíz. Los resultados obtenidos muestran que el algoritmo C&B tiene la capacidad de probar la optimalidad de la solución encontrada por las instancias hasta 25 clientes, y alcanza la mejor solución conocida para todas las instancias hasta 35 clientes. Sin embargo, el número de BKS encontradas para instancias de mayor tamaño decrece significativamente, y el tiempo límite (de dos a cinco horas) es alcanzado por todas las instancias con al menos 45 clientes. Los algoritmos MGA e ILS son más eficientes, dado que tienen la capacidad de obtener la mejor solución, respectivamente, para 15.9 y 15.7 de 20 instancias en promedio. Por lo tanto, el mejor comportamiento es alcanzado por el HA, el cual obtiene la mejor solución para 10.3 instancias de 20 en promedio. Adicionalmente el HA resulta ser el algoritmo más estable, en la medida en la que obtiene los mejores resultados considerando los valores promedio alcanzados a través de las 10 corridas ejecutadas para las instancias consideradas en este estudio. El trabajo futuro estará orientado a desarrollar métodos exactos que tengan el HA embebido en un algoritmo de *Branch-and-Cut* con el fin de resolver hasta optimalidad probada las instancias con un número mayor de clientes.

Capítulo 6

Algoritmo genético especializado para el problema de enrutamiento de vehículos con flota heterogénea y restricciones de empaquetamiento bidimensional minimizando el consumo de combustible 2L-FHFVRP

6.1. Introducción

El problema integrado de enrutamiento de vehículos y carga de mercancías considerando reducir el consumo de combustible, busca encontrar el conjunto de rutas de atención de la demanda de los clientes, argumentando que el consumo de combustible está directamente relacionado con el peso de la carga durante los trayectos que componen las rutas. En este estudio se integra el problema de enrutamiento de vehículos con flota heterogénea y el

problema de carga bidimensional, teniendo como objetivo reducir el consumo de combustible con base en el peso de la carga para cada trayecto 2L-FHFVRP.

Reducir el consumo de combustible pensando en el impacto ambiental asociado, es una variante del VRP clásico que ha ganado bastante atención en la última década, el objetivo de este problema es diseñar las rutas de entrega para satisfacer las demandas de los clientes con el menor consumo de combustible posible, el cual depende de las distancias de los trayectos, los vehículos asignados, el patrón de carga/descarga y el peso de la carga.

En la literatura del problema de ruteo de vehículos, los algoritmos aproximados han tenido un gran éxito, se resaltan los algoritmos evolutivos, los cuales aparecen en trabajos previos con una estructura bastante sofisticada obteniendo excelentes resultados, pero que son difíciles de implementar y adaptar a otras variantes como la aquí propuesta. En este estudio se presenta un algoritmo genético especializado para resolver el diseño de rutas, conservando su principal característica, su fácil implementación. Mientras que, la restricción de carga de mercancías es validada a través de un algoritmo GRASP, el cual ha tenido gran acogida para los problemas de empaquetamiento. Con el fin de validar el desempeño de la metodología propuesta, se presenta un estudio computacional que utiliza los diferentes casos de prueba de la literatura especializada, permitiendo ilustrar las economías alcanzadas en consumo de combustible. La metodología sugerida se adapta a la versión de minimizar únicamente la distancia total recorrida para el atendimento de los clientes (sin consumo de combustible) y se compara con los mejores trabajos presentados en la literatura, los resultados computacionales muestran que la metodología logra adaptarse suficientemente a esta versión, y en algunos casos de prueba logra encontrar mejores soluciones. Como trabajos futuros, se propone adaptar la metodología para considerar el problema de carga de mercancía tridimensional de forma tal que se adapte a condiciones más reales de la industria.

Principales Contribuciones

1. Se propone un algoritmo genético elitista de fácil implementación que resuelve de forma eficiente el problema enrutamiento de vehículos con flota heterogénea y restricciones de carga bidimensional buscando minimizar el consumo de combustible.

2. Se extiende la codificación propuesta por [Liu et al. \(2009\)](#) para representar las características de la carga y descarga secuencial de la mercancía, manteniendo las bondades de la codificación a la hora de cálculos y búsquedas de rutas.
3. Se adapta el concepto de espacios máximos para resolver los problemas de empaquetamiento bidimensionales con restricciones de carga y descarga secuencial.

6.2. Marco Referencial

La entrega de mercancías a clientes geográficamente dispersos juega un papel crucial en la gestión de la distribución y la logística, ya que representa una proporción considerable de los gastos de funcionamiento de las empresas, principalmente debido al alto consumo y costo de combustibles para soportar toda la operación. El problema de entregas al cual se ha puesto mayor atención se conoce como el problema de enrutamiento de vehículos (VRP), que describe la distribución de las mercancías desde un depósito principal a un conjunto de clientes a través de vehículos. El VRP se introdujo por primera vez por [Dantzig and Ramser \(1959\)](#), el problema trabajado por estos abordaba el envío de los camiones de reparto de gasolina entre un terminal de gráneles y un gran número de estaciones de servicio, desde la publicación de este trabajo un gran número de consideraciones han sido tomadas en cuenta para representar otros problemas de la industria. Originalmente el problema ha considerado minimizar únicamente las distancias totales recorridas, pero este objetivo ha sido demostrado por diferentes estudios que no está alineado a las preocupaciones de las empresas en la actualidad. Una de las variantes más estudiada en la última década es la de reducir el consumo de combustible, dado que este representa una disminución de costos importantes para la empresa y además comienza a resolver las preocupaciones de tener una operación más limpia, es decir, generar el menor impacto ambiental posible [Toth and Vigo \(2014\)](#). Es sabido que el consumo de combustible está directamente relacionado con las emisiones de gases de efecto invernadero (GHG), en especial las emisiones de CO₂ [Demir et al. \(2014\)](#). Por otro lado, debido a los avances en sistemas de cómputo más sofisticados y el desarrollo de técnicas de optimización más potentes se ha conseguido resolver problemas mucho más cercanos a las

condiciones reales de la industria, un ejemplo, es la solución del problema integrado de carga de mercancías y enrutamiento de vehículos, los cuales habían sido tratados por separado por más de 50 años. En la mayoría de estos trabajos, la carga de las mercancías era simplificada de manera que la forma geométrica de la carga era ignorada y sólo se consideraba una dimensión (volumétrica por lo general). En la práctica, los administradores logísticos deben solucionar problemas de transporte y carga de mercancías al mismo tiempo, sobre todo si la carga no es trivial (un solo tipo de producto, por lo general granel o líquido). Por ejemplo, cuando un elemento se apila en otros artículos, la estabilidad del apilamiento, la fragilidad y la facilidad de descarga para los operarios deben ser consideradas cuidadosamente. Por lo tanto, las soluciones de estos problemas sin integración resultan ser poco prácticas debido a las operaciones reales de carga y descarga de las mercancías.

Problemas integrados de ruteo y carga de vehículos como el 2L-CVRP y el 3L-CVRP han sido estudiados y resueltos de manera satisfactoria [Iori et al. \(2007\)](#) y [Gendreau et al. \(2006\)](#). La mayoría de los trabajos previos no consideran una flota heterogénea de vehículos, una flota homogénea de vehículos en la práctica es poco común. Una flota mixta de vehículos ofrece la flexibilidad para diseñar un plan de distribución más rentable [Subramanian et al. \(2013\)](#) y obedece a las actualizaciones del parque automotor en el tiempo.

Este estudio se realiza sobre una extensión práctica del VRP, en este problema, el depósito tiene una flota de vehículos heterogénea, las demandas de los clientes están conformadas por un conjunto de piezas rectangulares que representan objetos no apilables en la práctica, con largos, anchos y pesos conocidos. Se busca entonces seleccionar un conjunto de vehículos y determinar la ruta de cada vehículo para servir a todos los clientes, teniendo como objetivo minimizar el consumo de combustible de las rutas sumado a los costos fijos y variables de los vehículos seleccionados.

El consumo de combustible surge como un factor sobre la suma de las aristas que conforman un viaje realizado por el vehículo, relacionando en cada arista la carga con la que el vehículo viaja a través de ella y la capacidad de carga del vehículo. Mientras que los costos fijos se asocian a la selección de los vehículos y los costos variables dependen de las aristas recorridas

(distancia entre los clientes) para cubrir la ruta con el tipo de vehículo seleccionado. Los tipos de vehículos son heterogéneos en cuanto a su capacidad de carga, espacio de carga, costos fijos y costos variables. El número de ejemplares de cada tipo de vehículo es ilimitado. Una ruta se considera factible sólo si hay un plan de carga (descarga) viable para las piezas exigidas por los clientes en esta ruta, es decir, la descarga de mercancía de un cliente al llegar a su destino debe hacerse sin realizar operaciones de reordenamiento de las cajas restantes, esto en la literatura se llama restricción de carga secuencial, LIFO o *multi-drop*.

El 2L-FHFVRP también es de gran interés en la investigación teórica. Dado que es una combinación de dos problemas NP-duros bien conocidos, el problema de ruteo de vehículos con flota vehículos heterogénea (HFVRP) y el problema de empaquetamiento bidimensional 2D-BPP (por sus siglas en inglés *Two-dimensional Bin Packing Problem*). Iori et al. (2007) muestra que el problema integrado es por lo menos igual de complejo que uno de estos por separado, es decir, el 2L-FHFVRP también es NP-duro.

Para resolver este problema, se presenta un algoritmo genético especializado tipo elitista Chu and Beasley (1997). La representación de un cromosoma (solución) se adaptó y extendió de la estructura de datos propuesta por Liu et al. (2009). La estructura de datos consiste en el uso de un arreglo para registrar el orden de atención de los clientes y utilizar un grafo dirigido para fabricar el conjunto de posibles rutas dado un orden de atención.

La población inicial utiliza la solución heurística asociada al problema TSP de los nodos cliente originales. La selección se realiza por torneo, los dos cromosomas ganadores son recombinados a través de cinco operadores clásicos. Posteriormente, a cada uno de estos elementos se les calcula su función de adaptación la cual incluye la validación de las restricciones de empaquetamiento bidimensional, que penaliza los descendientes que no cumplen con estas. Por último, se presentan los nuevos individuos a la población y se regresa de nuevo a etapa de selección a no ser que el criterio de máximo número de generaciones sea ya alcanzado. La estructura elitista del algoritmo exige mantener dos poblaciones durante el proceso de optimización.

En particular, el validador de las restricciones de empaquetamiento bidimensional es un

algoritmo con una estructura tipo GRASP, que en su fase de construcción utiliza los espacios máximos para representar las piezas empacadas y los mayores espacios libres restantes, escoge pseudo-aleatoriamente entre criterios como agrupar las piezas por clientes o llevar todas las piezas hacia la parte posterior del vehículo. Por último, en su fase de mejoría, realiza movimientos de vaciado de piezas y llenado mediante criterios como la pieza que mejor encaja. La metodología propuesta, al permitir soluciones infactibles, logra diferenciarse entre los trabajos previos que sólo navegan en espacios de búsqueda factibles.

La codificación propuesta permite encapsular las asignaciones de los tipos de vehículos requeridos por el cromosoma dentro del grafo dirigido y desacopla la metaheurística de empaquetamiento llamándola únicamente cuando se tiene una solución del problema de ruta más corta (el cual define las secuencias de clientes a visitar). Para demostrar la eficacia de la metodología propuesta, se resolvieron todas las instancias de prueba presentadas en la literatura, logrando alcanzar soluciones en tiempos de cómputo razonables. Además, el algoritmo genético fue comparado con trabajos previos publicados para el problema 2L-HFVRP, los resultados demuestran que el algoritmo propuesto tiene un comportamiento aceptable, con relación a los mejores trabajos publicados en la literatura.

El resto de este capítulo está organizado de la siguiente manera. En la Sección 6.3 se describe formalmente el problema. La Sección 6.4 proporciona una visión general de la literatura del problema. En la Sección 6.5 la metodología propuesta es presentada detallando la estructura de datos y los algoritmos de optimización. El estudio computacional realizado es presentado en la Sección 6.6. Finalmente, la Sección 6.7 ofrece las conclusiones y trabajos futuros alrededor de este estudio.

6.3. Descripción del Problema

El problema de enrutamiento de vehículos con flota heterogénea y restricciones de empaquetamiento bidimensional minimizando el consumo de combustible, consiste en distribuir desde un depósito, la demanda de varios clientes que se encuentran geográficamente

dispersos, buscando minimizar el consumo de combustible sumado a los costos fijos y variables. Considerando la capacidad de carga o peso máximo que soporta cada uno de los vehículos que se empleen, y ubicando la demanda (ítems) de tal manera que, al visitar los clientes, no sea necesario reacomodar la carga correspondiente a los clientes posteriores.

Formalmente, este problema puede ser descrito de la siguiente manera. El 2L-FHFVRP es definido como un grafo completo $G = (V, E)$ donde $V = \{0, 1, 2, \dots, n\}$ es un conjunto de vértices (siendo 0 el depósito y los vértices $1, 2, \dots, n$ los clientes) y $E = \{e_{ij} : i, j \in V\}$ el conjunto de aristas. Para cada $e_{ij} \in E$, una distancia d_{ij} ($d_{ij} = 0$) es asociada. Una flota de M diferentes tipos de vehículos es ubicada en el depósito y un número muy grande de ejemplares por cada tipo es asumido. Cada vehículo t tiene asociada una capacidad Q_t , un costo fijo F_t , un costo variable V_t , una longitud L_t , un ancho A_t y un peso W_t (donde $t = 1, 2, \dots, M$). Se asume que un vehículo con mayor capacidad tiene mayores costos, y presenta mayor consumo de combustible, de esta forma si $Q_1 \leq Q_2 \leq \dots \leq Q_M$, entonces $F_1 \leq F_2 \leq \dots \leq F_M$ y $V_1 \leq V_2 \leq \dots \leq V_M$. Los costos variables de transporte por cada arista $e_{ij} \in E$ para el vehículo t son $C_{ijt} = V_t \cdot d_{ij}$. Por lo tanto, los costos de transporte (fijos más variables) de una ruta r para el vehículo tipo t son $C_r = F_t + \sum_{(i,j) \in R} C_{ijt}$, donde R representa el conjunto de aristas que conforman la ruta r ($R = \{(0, i), (i, j), \dots, (k, 0)\}$). El costo de consumo de combustible para cada arco (i, j) incurrido por un vehículo tipo t transportando una carga con peso q es $C_{fc} = fc(q) \cdot C_{ijt}$. Es importante anotar que, para el mismo arco, el consumo de combustible difiere dependiendo de la cantidad de carga transportada y el vehículo que la transporta. Como en [Zhang et al. \(2015\)](#) $fc(q)$ es definido como $\frac{q}{Q_t}$, siendo q la sumatoria de los pesos de las demandas de los clientes que están siendo transportadas. De esta forma, en la Ecuación 6.1 se presenta la función objetivo del problema, la cual consiste en los costos de transporte más los costos de consumo de combustible.

$$Z = C_r + C_{fc} \quad (6.1)$$

Por otro lado, cada cliente c (donde, $c = 1, 2, \dots, n$) requiere un conjunto de B_c piezas rectangulares con un peso total w_c , denotadas como I_c y su peso total es igual a D_c . Cada ítem

$I_{cb} \in B_c (b = 1, 2, \dots, |B_c|)$ tiene un largo específico l_{cb} y un ancho w_{cb} . En el 2L-FHFVRP, las rutas de los vehículos y el empaquetamiento de las mercancías deben satisfacer las siguientes restricciones:

- Todas las rutas deben comenzar y terminar en el depósito.
- Todas las piezas de un cliente deben estar empacadas en el mismo vehículo, es decir, los clientes deben ser atendidos una única vez.
- Todas las piezas deben ser cargadas con sus lados paralelos a las paredes del vehículo.
- La capacidad, el largo y el ancho del vehículo no pueden ser excedidos por la carga.
- La carga no puede traslaparse entre sí.
- Cada ítem debe ser posicionado dentro del vehículo, de forma tal que, al arribar a su destino, este podrá ser descargado sin la necesidad de reorganizar los ítems de otros clientes.

De esta forma, el 2L-FHFVRP busca seleccionar un conjunto de vehículos para realizar las rutas dada la asignación de clientes a estas, con el fin de minimizar la Ecuación 6.1, sujeto al conjunto de restricciones anteriormente enunciado.

6.4. Estado del Arte

Un punto de partida del problema de rutas de vehículos considerando restricciones de empaquetamiento, es la revisión del estado del arte realizada por Iori (2013), la cual es mejorada por Pollaris et al. (2015). Luego de estudiar los trabajos previos, es necesario resaltar la relevancia de resolver el problema de rutas de vehículos integrado con la carga de mercancías, buscando optimizar objetivos reales de la industria, para esto, los autores recomiendan leer Côté et al. (2017).

Hasta la fecha de publicación de este trabajo, se desconocen estudios previos en el 2L-FHFVRP, por esto, la revisión bibliográfica se enfocó en el 2L-HFVRP, el cual sólo difiere

en la función objetivo a tratar. El primer trabajo reportado en la literatura presenta un algoritmo de recocido simulado, el cual obtiene resultados razonables, y es propuesto por [Leung et al. \(2013\)](#). En este trabajo se resuelven también el 2L-CVRP y el HFVRP de manera satisfactoria. Por otro lado, los trabajos propuestos por [Dominguez et al. \(2016\)](#) y [Domínguez Rivero et al. \(2016\)](#), son algoritmos tipo ILS que logran superar los mejores resultados reportados en la literatura. Estos trabajos son relevantes, dado que resaltan la importancia de considerar condiciones reales de la operación de transporte de mercancías, para esto, los autores realizan un estudio sobre diferentes escenarios evaluando características como: la rotación de ítems, carga/descarga secuencial, y el uso de una flota heterogénea de vehículos.

Considerar el consumo de combustible dentro de la función objetivo permite generar un conjunto de rutas en las cuales, se realizan largos trayectos con los vehículos menos cargados (ver Figura 6.1). Esta práctica en la literatura se conoce como problemas de rutas de vehículos con impacto ambiental, este es formalizado en [Suzuki \(2011\)](#) y [Bektaş and Laporte \(2011\)](#). Es [Xiao et al. \(2012\)](#) quienes por primera vez relacionan el peso de la carga y la distancia recorrida, con el consumo de combustible dentro de la función de costos. Luego de esto, [Zhang et al. \(2015\)](#) simplifican esta relación, estableciendo así una función objetivo que busca minimizar los costos de transporte sumado a los costos de consumo de combustible.

Revisando trabajos que integren rutas de vehículos y empaquetamiento de mercancías bajo condiciones prácticas, se puede encontrar que asumir una demanda o tiempos de viaje determinísticos son suposiciones fuertes que afectan el desempeño de las soluciones en aplicaciones reales. Cabe resaltar entonces los trabajos propuestos por [Guimarans et al. \(2016\)](#) y [Guimarans et al. \(2018\)](#), los cuales incorporan características de incertidumbre dentro de su análisis, acercándolos aún más, a problemas reales de rutas de vehículos. Estudios recientes vinculan las características de incertidumbre en tiempos de viaje y demanda de la carga, para combinarlas con las distintas capacidades de la flota, y establecer una función objetivo cercana a las preocupaciones actuales de los transportistas, como en el trabajo de [Micheli and Mantella \(2018\)](#).

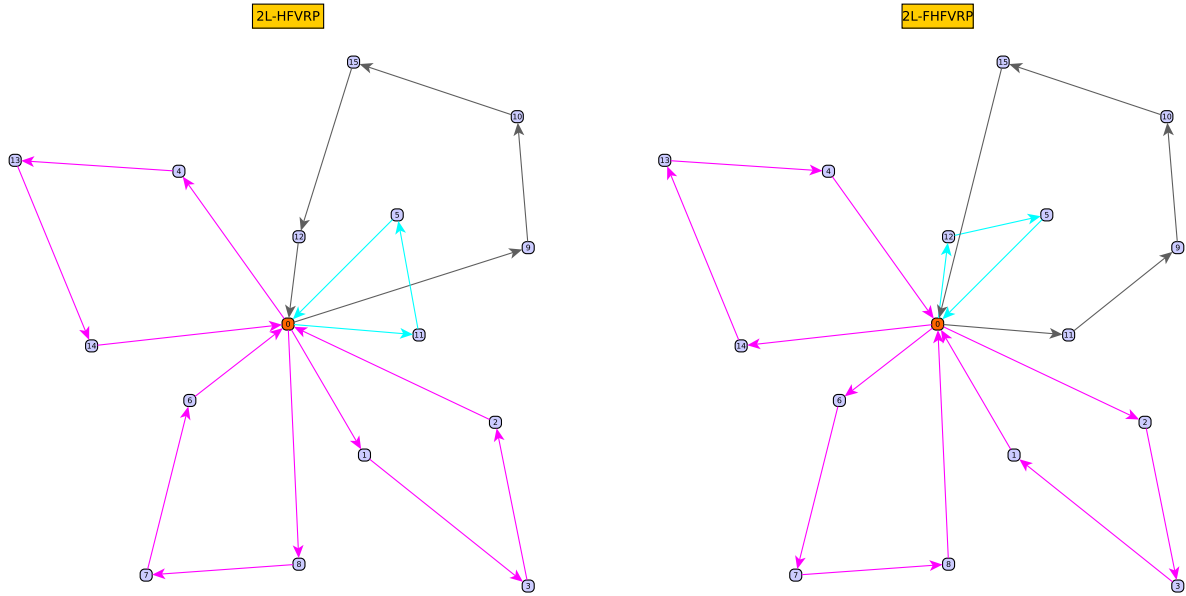


Figura 6.1: Incidencia del impacto ambiental, del 2L-HFVRP al 2L-FHFVRP.

La variante ambiental de este estudio es conocida como Problema de Enrutamiento de Vehículos Minimizando Consumo de Combustible, la cual busca reducir los costos fijos y variables asociados a las rutas, sumado a los generados por el consumo de combustible, esta versión se alinea con las preocupaciones de las empresas actuales. Por otro lado, en la literatura se encuentran estudios del problema considerando restricciones de empaquetamiento tridimensionales, el trabajo más relevante en esta dirección es el algoritmo de búsqueda local evolutiva presentado por [Zhang et al. \(2015\)](#), el cual logra resolver de forma satisfactoria el problema combinado de rutas y empaquetamiento minimizando el consumo de combustible. Este estudio busca ampliar el trabajo propuesto por [Zhang et al. \(2015\)](#) al considerar la presencia de una flota mixta de vehículos en contextos de carga no apilable.

6.5. Metodología de Solución

En este estudio se propone un algoritmo metaheurístico, intentando explotar la facilidad de implementación de estas metodologías. Se adapta la estructura de un GA presentado en [Chu and Beasley \(1997\)](#), y se incorpora un algoritmo GRASP basado en diferentes trabajos

publicados en el área de corte y empaquetamiento. Cabe resaltar que, la base de ambos algoritmos es la codificación utilizada, para el GA se adaptó el procedimiento de *split*, mientras que para el GRASP se adecuó el concepto de espacio residuales máximos.

Codificación

La codificación planteada por [Prins \(2004\)](#) consiste en un vector que representa el orden de atención de todos los clientes, esta se ayuda de un grafo dirigido auxiliar, que propone crear todos los posibles viajes siguiendo el orden establecido, garantizando la restricción de capacidad del vehículo disponible. Dado que este grafo comprende todas las rutas factibles dado un orden, se debe entonces seleccionar el mejor conjunto de rutas que visiten todos los clientes a menor costo, para esto es necesario resolver un problema de ruta más corta. Esta codificación es extendida por [Liu et al. \(2009\)](#) permitiendo considerar una flota heterogénea de vehículos. En este trabajo, se adapta para representar también las restricciones de empaquetamiento, buscando en la creación y actualización del dígrafo, realizar el filtrado de los viajes que no son viables por empaquetamiento secuencial en los diferentes vehículos presentes en la flota.

De esta forma entonces, la construcción del dígrafo consistirá en la creación de los arcos correspondientes a los viajes que se podrían realizar considerando únicamente la restricción de capacidad (peso soportado) de los distintos vehículos. Dado que los costos fijos y variables de transporte se relacionan directamente con la capacidad del vehículo, no es necesario crear los mismos viajes (arcos) para vehículos de mayor porte si ya un vehículo menor puede realizar este viaje. Por lo tanto, no considerar las restricciones de empaquetamiento en la creación, requiere entonces verificar si la solución del problema de ruta más corta no contiene viajes infactibles. Para esto, se utilizará un algoritmo GRASP, el cual será descrito más adelante. Así, cada uno de los arcos de la solución deberá ser validado, en caso de no ser empaquetable, se prueban vehículos de mayor tamaño hasta agotar la flota disponible, si no es posible encontrar un vehículo para transportar la mercancía, este arco debe ser eliminado del dígrafo, y se debe resolver de nuevo un problema de ruta más corta.

Es importante resaltar que el grafo auxiliar resultante es un grafo dirigido acíclico (ADG), esto permite utilizar la versión del algoritmo de Bellman-Ford para ADG [Ahuja et al. \(1988\)](#), para calcular la ruta más corta de manera eficiente. De esta forma, se actualizan primero todos los arcos infactibles por empaquetamiento y con el grafo actualizado se llama al algoritmo de Bellman-Ford. Este esquema de codificación ofrece gran flexibilidad para tratar múltiples variantes del problema, la selección de arcos que constituirán el dígrafo y la información que se asociará a cada uno de estos, brinda la posibilidad de representar variantes que permitan apilamiento (empaquetamiento tridimensional), múltiples depósitos, flota homogénea o heterogénea, ventanas de tiempo, entre otras.

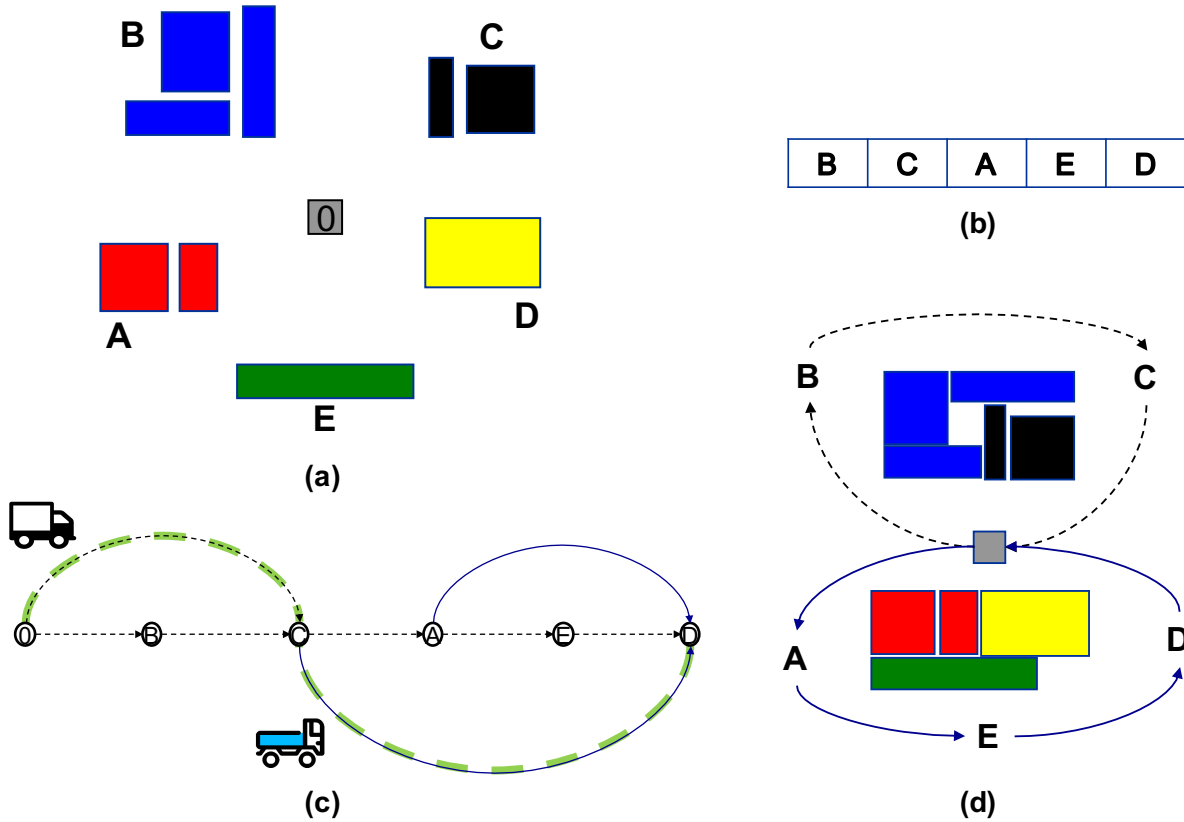


Figura 6.2: Codificación y decodificación de solución. **a.** Conjunto de ubicaciones y piezas demandadas de los clientes. **b.** Ejemplo de orden de atención. **c.** Grafo auxiliar del ejemplo y ruta más corta. **d.** Conjunto de rutas y empaquetamiento dentro de estas.

El objetivo principal de este trabajo es enriquecer los problemas de rutas de vehículos y empaquetamiento de mercancías, incluyendo la minimización del consumo de combustible.

En términos de codificación, el impacto ambiental se verá reflejado en el costo de los arcos del grafo auxiliar. La Figura 6.2.a ilustra un caso de prueba con cinco clientes dispersos, la codificación propuesta requiere entonces almacenar en un arreglo el orden de atención de estos, la Figura 6.2.b muestra un ejemplo de atención de estos cinco clientes. Para encontrar el conjunto de rutas factibles es necesario crear el grafo auxiliar, usando los distintos vehículos disponibles, la Figura 6.2.c ilustra un ejemplo de grafo para dos tipos de vehículo (V1 y V2), un arco punteado representa un viaje realizado por un vehículo tipo V1, mientras los arcos azules continuos representan los viajes realizados en vehículos tipo V2. Por otro lado, en la Figura 6.2.c, el camino punteado verde corresponde a la ruta más corta, resultante de ejecutar el algoritmo de Bellman-Ford utilizando como costo de los arcos, el costo de cada viaje dado por la Ecuación 6.1. Esta solución verde punteada, es la que entrará al esquema iterativo de verificación de la restricción de empaquetamiento para cada viaje. Cuando todos los arcos resultantes sean válidos, finalmente se obtendrá una solución factible dado el orden de atención, así como se muestra en la Figura 6.2.d, donde se realizan dos viajes, el primero en el vehículo V1 atendiendo los clientes B y C, mientras el segundo en el vehículo V2 visitando a los clientes A, E y D.

6.5.1. Algoritmo GRASP de empaquetamiento bidimensional

En este trabajo se adaptada un algoritmo GRASP presentado en [Martínez et al. \(2015\)](#), el cual en la fase de construcción permite obtener soluciones factibles a través del control del posicionamiento de las piezas dentro del vehículo, satisfaciendo así la restricción de carga secuencial. El algoritmo GRASP fue desarrollado por [Feo and Resende \(1989\)](#) para resolver problemas de optimización combinatoria difíciles. El GRASP es un procedimiento iterativo que combina una fase constructiva y una fase de mejora. En la fase constructiva una solución es construida paso a paso, agregando elementos a la solución. La fase debe ser iterativa, golosa, aleatoria y adaptativa. La estrategia de búsqueda propuesta en este algoritmo permite en la etapa de construcción aleatorizar la elección de las piezas a ubicar en cada espacio vacío. Mientras que en la fase de búsqueda local se realizan movimientos de vaciado y relleno.

El Algoritmo 9 propuesto comienza por seleccionar uno de los espacios libres (s'), seguido de esto se genera la lista de capas de piezas (L) que caben en el espacio s' . Luego esta lista L es reducida a la Lista Restricta de Candidatos (RCL), dentro de esta lista reducida es seleccionada aleatoriamente uno de sus elementos (C). Esta capa C es ubicada generando así el patrón P y obligando a actualizar las listas de espacios vacíos y la lista de piezas restantes (S , B_j , y B , respectivamente). Si al terminar de empacar las piezas de un cliente existen espacios libres restantes y aún hay clientes por empacar, entonces la lista de espacios vacíos debe ser actualizada por cambio de cliente, en esta se deben eliminar los espacios que violen las restricciones de carga secuencial. Cuando hemos terminado el intento de asignar todas las piezas demandadas por los clientes de la ruta, analizamos si la función objetivo del patrón obtenido es de buena calidad, es decir, si esta solución construida es promisorio. Utilizando la función *MejorarPatrónConstruido*(P), se removerán aleatoriamente piezas y se intentará llenar de nuevo el vehículo eliminando del esquema de búsqueda las elecciones aleatorias. Por último, se verifica si todas las piezas demandadas han sido empacadas, retornando verdadero si es factible. En caso de que se agoten las iteraciones y en ninguno intento se ha conseguido empacar la totalidad de piezas, el algoritmo retornará infactible.

Los detalles de la fase de construcción y aleatorización del algoritmo de empaquetamiento, se encuentran en (Martínez et al., 2015). El algoritmo constructivo de este trabajo es basado en la fase constructiva propuesta por Escobar et al. (2015). Dado que ambos trabajos comparten la característica de carga dividida en clientes, la cual requiere un tratamiento especial a las estructuras de actualización de los espacios restantes y utilizables (vacíos). Esta parte es laboriosa dado que los espacios vacíos deben ser verificados para garantizar la carga secuencial, esto implica que algunos deben ser eliminados o recortados. El método constructivo utiliza los espacios máximos vacíos para representar el área utilizable del vehículo. En este caso, cada pieza seleccionada es empacada en un espacio vacío y esto genera a lo sumo dos nuevos espacios máximos (ver Figura 3). El algoritmo constructivo usa una lista S de los espacios máximos y una lista B_j de las cajas del cliente j que aún deben ser empacadas. Los pasos 0 – 4 resumen el algoritmo constructivo.

Paso 0: Inicialización. Es creada una lista S de los espacios máximos vacíos e iniciada con

Algorithm 9 Validador de Empaquetamiento Tipo GRASP

Data: Route, Vehicle, Customers, Max.Iterations

Result: feasible or unfeasible

```
1 for  $i \leftarrow 1 \dots \text{Max\_Iterations}$  do
2   Lista  $S = \text{Vehicle.Dimensions}$ 
3   Lista  $B = \text{Customers.Pieces}$ 
4   for  $i \leftarrow 1 \dots \text{Max\_Iterations}$  do
5     Lista  $B_j = \text{Customers}(j).Pieces$ 
6     while  $B_j \neq \emptyset$  and  $S \neq \emptyset$  do
7       Espacio  $s' = \text{SeleccionarEspacio}(S)$ 
8       Lista  $L = \text{GenerarListaCapas}(s', B_j)$ 
9       Lista  $RCL = \text{ConstruirRCL}(L, \delta)$ 
10      Capa  $C = \text{SeleccionarCapaAleatoriamente}(RCL)$ 
11      Patrón  $P = \text{LocalizarCapa}(C, s', P)$ 
12      Lista  $S = \text{ActualizarListaEspaciosMáximos}(C, S)$ 
13      Listas  $B_j, B = \text{ActualizarListaCajasRestantes}(C, B_j, B)$ 
14    end
15    Lista  $S = \text{ActualizarListaEspaciosCambioCliente}(S)$ 
16    if  $\text{EsPromisorio}(P)$  then
17      Patrón  $P = \text{MejorarPatrónConstruido}(P)$ 
18    end
19    if  $B = \emptyset$  then
20      return true
21    end
22  end
23 end
24 return false
```

el espacio (área) del vehículo a realizar la ruta. Sean las listas B_1, B_2, \dots, B_n , los conjuntos de ítems de cada cliente a ser atendido en la ruta y la lista B el conjunto de todas las piezas demandadas.

Paso 1: Escoger un espacio máximo de S . Dado que la lista S contiene los rectángulos vacíos más grandes disponibles y que existen varios candidatos (piezas a empacar), se debe determinar un mecanismo de selección con base en algún criterio de calidad o estrategia de empaquetamiento, en este trabajo se proponen dos criterios de selección: escoger el espacio máximo con mínima distancia al fondo del vehículo y escoger el espacio máximo con mayor área. Además de esto, del espacio elegido es seleccionada la esquina trasera más cercana a una de las esquinas traseras del vehículo, como punto de anclaje (o referencia) a la hora de ubicar las piezas en el espacio vacío.

Paso 2: Escoger las piezas a empacar. Una vez que el espacio máximo s' ha sido seleccionado, se debe tomar de la lista ordenada B_i (siendo i el cliente actual a empacar) cada caja k que quepa dentro de s' . Si existen varios ejemplares de la caja k , se debe generar cada una de las posibles capas. Esto consiste en empacar las piezas en arreglos de columnas o filas combinando los diferentes ejes. Como en [Escobar-Falcón et al. \(2016\)](#), dos criterios son considerados para seleccionar una de las configuraciones de piezas. Escoger la capa de piezas que produce el mayor incremento en la función objetivo (área máxima) o escoger la capa de piezas que mejor encajan en el espacio máximo (mejor ajuste). Este último es un criterio en el cual se computan las distancias entre las caras de la capa de piezas y el espacio máximo, para luego ordenar las distancias de cada configuración en orden no-decreciente y seleccionar la configuración con menor distancia (la que mejor se ajusta al espacio).

Paso 3: Actualizar la lista S . A menos que la pieza (o la capa) quepa exactamente en el espacio s' , el empacar producirá nuevos espacios máximos vacíos que reemplazarán a s' en la lista S . Por otra parte, como los espacios máximos no son disjuntos, las piezas empacadas pueden interceptarse con otros espacios máximos los cuales pueden ser reducidos o eliminados. La lista B es también actualizada y los espacios máximos que no puedan ubicar ninguna de las piezas que aún restan por empacar deberán ser eliminados de S . Si aún restan piezas de

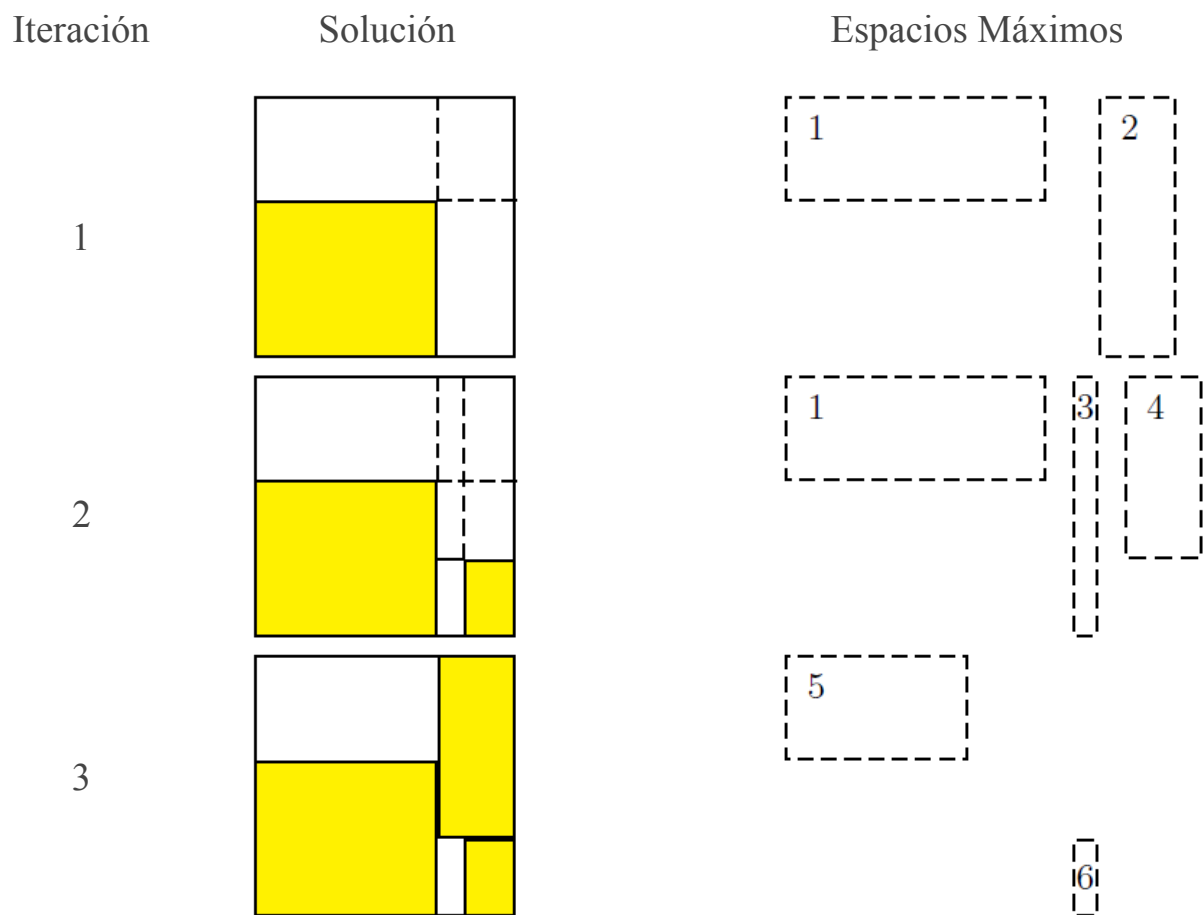


Figura 6.3: Evolución de los espacios máximos en tres iteraciones de la fase constructiva. Adaptado de [Parreno et al. \(2010\)](#)

un cliente por ser empacadas se debe volver al Paso 1.

Paso 4: Actualizar la lista S para el siguiente cliente en la ruta. Cuando el cliente actual ha sido empacado, los espacios máximos deben ser actualizados para garantizar las restricciones de carga secuencial, es decir, serán removidos de la lista todos los espacios máximos que son completamente invisibles desde la puerta del vehículo. También deben ser actualizados (modificados) los espacios máximos que tienen una parte visible y otra invisible. Si S o B son vacíos se da por terminada esta fase.

Debido a que una construcción voraz no permitiría una buena exploración sobre el espacio de búsqueda, un alto grado de aleatorización es introducido en el Paso 2. Por lo que, para cada tipo de pieza es construida una capa, de acuerdo con el criterio de selección (área máxima o mejor ajuste). Cada una de estas es llamada como: configuración o candidato. Con todo el conjunto de posibles capas es construida una lista restringida de candidatos y es seleccionado uno de estos al azar. El parámetro $\alpha \in [0, 1]$ controla el tamaño de la lista de candidatos. El desempeño del algoritmo constructivo depende principalmente de este parámetro, por eso, en este trabajo se calibró de manera reactiva como lo propone [Camacho et al. \(2018\)](#), permitiendo así diferentes tamaños de lista que se ajusten de acuerdo con su desempeño histórico. La fase de mejora consiste en realizar el proceso constructivo anteriormente enunciado, eliminando todas las características aleatorias, luego de haber eliminado $K\%$ de las cajas empaquetadas. El porcentaje de cajas y las cajas para retirar de la solución se realiza como se recomienda en [Escobar et al. \(2015\)](#).

6.5.2. Algoritmo GA-GRASP

La metodología de solución de este estudio adaptó un Algoritmo Genético tipo *Chu-Beasley*, donde se utilizan operadores de recombinación sobre vectores de orden de atención de clientes, sobre los cuales se aplica el procedimiento de *split*, el cual ayuda a determinar la función objetivo, la adaptación del *split* a problema de este trabajo fue descrito en la Sección 4.1.

El Algoritmo 10 ilustra el esquema de optimización propuesto. En este la población inicial es parcialmente generada de forma aleatoria (Líneas 1-3), pero la mayor parte es construida por

las soluciones encontradas por el algoritmo basado en *Simulated Annealing*, en este utiliza como solución inicial el TSP obtenido por la heurística de Christofides (1976), aplicada únicamente a los clientes (Líneas 4-7). En total son generados $\left\lceil \frac{n}{2} \right\rceil$ individuos, siendo n el número de clientes que deben ser atendidos en cada caso.

Por otro lado, el esquema generacional del genético propuesto consiste en selección, recombinación y mutación. La selección se realiza por torneo (Línea 9), en este, se selecciona un par de padres. En la recombinación se emplean cinco operadores clásicos de la literatura (PMX, SJX, OX, CX y OBX), generando un hijo por cada operador (Línea 10).

Después de esto, comienza entonces la fase crítica del algoritmo, la cual requiere del cálculo de la función de adaptación (*fitness*) para cada hijo creado. Para esto, se adapta el *split* de forma tal que, se encuentre de forma eficiente el conjunto de rutas, para el orden de atendimento dado en el cromosoma. De esta manera, el primer paso consiste en crear el grafo dirigido (Línea 12), considerando los distintos vehículos disponibles en la flota y los pesos de los ítems, y teniendo como costo de los arcos la función objetivo (Ecuación 6.1) de los arcos que componen la ruta.

Luego de la creación del grafo, se debe resolver el problema de camino más corto sobre este, para identificar el conjunto de rutas que visitan todos los nodos de forma más económica, dado que el grafo resultante es un grafo dirigido acíclico (ADG), podemos utilizar la versión del algoritmo de Bellman-Ford para este tipo de grafos (Línea 14). Las rutas seleccionadas (arcos que componen el camino más corto) tienen la oportunidad de ser mutadas (Línea 16), la mutación de las rutas consiste encontrar la primera mejora a través del operador 2-opt clásico del TSP (Línea 17), las rutas mutadas afectan directamente el cromosoma, de esta forma, el orden de atendimento es corregido para que el individuo pueda transmitir esta información en las siguientes generaciones. Las rutas mutadas o no, pueden ser infactibles desde el punto de vista de empaquetamiento, por lo que es necesario validarlas con el GRASP (Línea 19). Cada vez que existe una infactibilidad esta es corregida en el grafo, eliminando el arco correspondiente (Línea 21), mientras existan infactibilidades, se debe seguir ejecutando el *split*.

Algorithm 10 Algoritmo Híbrido GA-GRASP

Data: depot, vehicles, customers, items, max_iterations, mutation_rate, diversity and seed

Result: Solution Routes

```
1 for  $i = 1 \dots \lfloor \frac{n}{6} \rfloor$  do
2   | population[i] = CreateRandomSolution(seed)
3 end
4 constructive_solution = ChristofidesTSP(customers)
5 for  $i = \lceil \frac{n}{6} \rceil \dots \lfloor \frac{n}{2} \rfloor$  do
6   | population[i] = SimulatedAnnealing(constructive_solution)
7 end
8 for  $k = 1 \dots Max\_Iterations$  do
9   | parents = Tournament(population)
10  | offspring = CrossoverPMX-SJX-OX-CX-OBX(parents)
11  for  $i = 1 \dots offspring.Size()$  do
12    | graph = CreateDigraph(offspring[i], depot, vehicles, customers, items)
13    while  $flag\_SP == false$  do
14      | shortest_path = Bellman-FordADG(graph)
15      for each route in shortest_path do
16        | if  $mutation\_rate > Rnd()$  then
17          | route = Mutate2OPT(route)
18        end
19        | flag = GRASP(route, items)
20        if  $flag == false$  then
21          | graph = DeleteUnfeasibleArc(route)
22          | flag_SP = true
23        end
24      end
25    end
26    if  $AcceptanceCriteria(population, diversity, Decode(shortest\_path))$  then
27      | UpdatePopulation(Decode(shortest_path))
28    end
29  end
30 end
31 return  $BestSolution(population)$ 
```

Por último, cada individuo factible se presenta a la población actual, si este cumple el grado

de diversidad y su función objetivo es mejor que al menos el peor individuo de la población, este entrará a reemplazarlo (Línea 25 y 26). Además de esto, se implementó el criterio de aspiración, esto significa que, aunque el individuo no sea diverso, si este tiene mejor función objetivo que el peor individuo del “grupo élite” de la población, este deberá ingresar al grupo élite, desplazando así al peor individuo de la población [Chu and Beasley \(1997\)](#). De esta forma, la estructura básica de un algoritmo elitista es conservada.

Dado que validar la restricción de empaquetamiento es un problema NP-Hard, es importante definir un esquema de factibilización de rutas rápido y eficiente. Para esto, como estrategias de aceleración se implementó un almacenamiento de los resultados ofrecidos por el GRASP, de esta manera, se consulta primero la tabla *hash* que guarda las validaciones ya hechas. Buscando eliminar el número de veces que el GRASP es llamado.

6.6. Experimentos computacionales y análisis de resultados

En esta sección se presentan los resultados alcanzados por la metodología propuesta, para esto se utilizó el conjunto de instancias clásicos publicados en la literatura. La batería de casos está dividida en cinco clases, cada una compuesta por 36 instancias. Debido a que no existen trabajos previos que aborden el problema de minimización de combustible, se analizó el empeoramiento en distancia de las rutas obtenidas, teniendo como objetivo minimizar el consumo de combustible, para esto se ejecutó la metodología propuesta sobre los casos y de igual manera se evaluó el comportamiento al eliminar el consumo de combustible. Siguiendo esta idea, se utilizaron los resultados obtenidos sin considerar el combustible para compararlos con trabajos previos de forma justa. De esta forma, se compara el desempeño la metodología y se demuestra que logra alcanzar mejores resultados para algunas instancias, lo que valida el correcto desarrollo del algoritmo. Todos los experimentos realizados en este estudio fueron ejecutados sobre una máquina con las siguientes especificaciones, un procesador *Intel® Core™ i5-4570* CPU de 3.20 GHz, una memoria RAM de 8GB y un sistema operativo Ubuntu Linux 16.04.03 de 64-bits. Todos algoritmos fueron implementaron en C++®.

Algoritmo	Parámetro	Valor*
GA	Número de generaciones	$\frac{n}{2 \log n}$
	Número de individuos	$\frac{n}{2}$
	Número de individuos generados aleatorios	$\frac{n}{6}$
	Tasa de mutación (%)	$\frac{n}{\log n}$
	Diversidad (genes)	$\frac{n}{2 \log n}$
GRASP	Número de iteraciones de empaquetamiento	$\frac{m}{2 \log m}$
	Porcentaje de cajas retiradas (%)	30

Tabla 6.1: Valores seleccionados para los parámetros. * Siendo n el número de clientes a rutear y m el número de cajas a ser empacadas en la ruta.

La Tabla 6.1 muestra que los valores de los parámetros del algoritmo propuesto están en su mayoría calibrados con relación a la información del problema, para esto se realizó un ajuste experimental de las relaciones para cada parámetro, utilizando los intervalos de valores recomendados en la literatura. El algoritmo genético requiere de cinco parámetros, cada uno de estos se logró calibrar con respecto al número de clientes a rutear. Por otro lado, el algoritmo GRASP solo requiere de dos parámetros debido a su naturaleza reactiva, que consigue calibrar su parámetro principal α , basado en el desempeño histórico para un número reducido de valores.

La Tabla 6.2 muestra los resultados computacionales obtenidos por la metodología propuesta, donde la columna Costo, es la función objetivo alcanzada por la mejor solución encontrada, la columna Tiempo de cómputo, es el tiempo computacional en segundos, requerido para alcanzar esta solución. Distancia es la suma de distancias de los arcos que conforman la solución. Mientras que, la columna GAP es la diferencia porcentual entre las distancias de la solución minimizando costos de combustible y la solución minimizando únicamente las distancias.

Id	Clase 1				Clase 2				Clase 3				Clase 4				Clase 5			
	Costo	Tiempo de Cómputo	Distancia	GAP (%)	Costo	Tiempo de Cómputo	Distancia	GAP (%)	Costo	Tiempo de Cómputo	Distancia	GAP (%)	Costo	Tiempo de Cómputo	Distancia	GAP (%)	Costo	Tiempo de Cómputo	Distancia	GAP (%)
1	780	<1	604	0	804	<1	647	3,7	824	<1	668	1,2	829	<1	666	0	780	<1	635	0
2	887	<1	698	0,5	975	<1	771	2,3	1043	<1	846	0,1	937	<1	735	0	887	<1	727	1,9
3	1034	<1	810	0,4	1082	<1	865	0	1070	<1	871	0	1022	<1	822	0	1034	<1	855	1
4	1007	<1	799	3,6	981	<1	755	0,4	981	<1	755	0,4	1000	<1	776	1,6	1007	<1	838	3,7
5	1086	<1	843	0,7	1182	<1	940	0	1165	<1	919	1,4	1142	<1	924	3	1086	<1	894	5,2
6	1145	<1	853	0	1215	<1	931	0	1215	<1	931	2,5	1298	<1	1012	0	1145	<1	886	0
7	4414	<1	3509	0	8851	<1	7587	2,3	7301	1	6052	0	7679	2	6641	1,5	4414	2	5945	0,9
8	4413	<1	3441	0	7431	<1	6096	0	8585	1	7426	0	8679	<1	7451	0	4413	4	4336	0,8
9	1498	<1	1116	1,7	1500	<1	1182	0,4	1490	<1	1174	0,4	1501	<1	1182	0,4	1498	<1	1192	1,7
10	7176	<1	5591	4	11261	1	9350	0	10376	2	8646	0	11485	3,7	10042	1,4	7176	9	7091	4,1
11	7136	<1	5351	0,9	11379	1	9355	0	11924	2	10092	0	12572	5,4	10703	1,1	7136	13	7038	0
12	2244	<1	1764	0,8	2296	<1	1823	0,3	2292	<1	1841	0,9	2287	<1	1816	0,5	2244	<1	1802	0,3
13	22251	<1	16692	0,9	37425	2	31794	2,2	36753	3	30907	4,7	38896	2,3	32619	2	22251	9	26745	2,4
14	13100	<1	10124	0,2	15085	<1	12032	0	16082	1	13336	0	15682	<1	12894	0	13100	1	12102	0
15	12920	<1	9988	0,9	15107	<1	12381	0	14938	<1	11989	0	17062	1,2	14248	0	12920	3	13351	0
16	1749	<1	1343	0,4	1806	<1	1400	0	1824	<1	1424	0,7	1801	<1	1382	0	1749	<1	1343	0,2
17	2379	<1	1827	0,1	2452	<1	1925	0	2418	<1	1837	0	2429	<1	1878	0	2379	<1	1894	0
18	4594	<1	3521	2,6	7260	3	6437	0,5	6991	4	6083	0	8187	16,6	7354	0,3	4594	14	5117	0
19	2137	<1	1637	1,3	6109	3	5312	1	6048	5	5412	1,3	6504	10,6	5727	0,6	2137	17	4077	2
20	2446	<1	1945	0	7097	26	6538	0,2	6747	60	6217	0,7	7459	97,1	6904	0	2446	178	5195	0,5
21	3539	<1	2713	0,7	11245	5	9914	0,5	12090	11	10687	0,4	11042	15,1	9613	0,8	3539	25	7736	1,2
22	3582	<1	2765	1,3	11247	5	10002	0,2	12283	11	10887	1,1	11504	20	10102	0,4	3582	40	8137	0,3
23	3539	<1	2713	0,7	11782	5	10342	0,8	10964	11	9719	0,2	11444	18,5	10157	0,6	3539	31	7589	0,8
24	3523	<1	2705	0,8	6819	1	5432	0,1	6303	1	5120	2,4	6288	2,3	4997	1,8	3523	2	4307	2,3
25	3958	<1	2972	0,4	15831	14	14196	0,2	14211	27	12659	0,2	15797	47,3	14271	0,8	3958	101	9097	1,2
26	5058	<1	3741	0,3	15762	6	13667	1,2	15565	15	13354	0,7	16367	27,2	14164	0,7	5058	52	9304	1
27	4039	<1	3034	0,2	7825	2	6210	0,3	8002	3	6482	0,8	7851	6,3	6396	1,4	4039	11	5691	1,2
28	6114	<1	4446	0,9	27734	31	24970	0	26203	58	23560	0	28516	102,6	25668	0,4	6114	228	18350	0,2
29	14545	<1	10957	1,4	28022	55	24182	0,1	25478	116	22077	0	27280	244,1	23534	0,6	14545	307	21425	0,1
30	5633	<1	4311	0,6	21571	17	19004	0,4	21301	34	18902	0,6	22363	69,4	19933	0,1	5633	113	13314	0,8
31	7722	<1	5924	0,5	26599	16	23530	0,5	27467	31	24099	0,3	28355	84,4	25150	0,2	7722	133	17463	0,3
32	7677	<1	5846	0,2	27738	20	24606	0,4	24942	44	22031	0,8	28337	81,9	24816	0,5	7677	123	16280	0,3
33	7805	<1	5952	0,1	26983	20	23770	0,2	26428	44	23364	0,5	28718	79,2	25457	0,7	7805	134	17357	1,3
34	5744	<1	4589	0,3	18066	14	16046	0,2	17228	40	15294	0,5	19662	74	17476	0,2	5744	104	11708	0,5
35	5372	<1	4164	0,1	11259	19	9851	0	11397	36	10025	0,1	12025	72,4	10637	0,1	5372	184	8574	0,2
36	3765	<1	3011	0	5806	20	5124	0	5802	37	5120	0	5543	61	4861	0	3765	141	4288	0
Promedio		<1		0,8		7,9		0,5		16,6		0,6		31,9		0,6		55		1

Tabla 6.2: Resultados obtenidos por el GA-GRASP

En la Tabla 6.2 se ilustra que la metodología propuesta logra resolver todas las instancias de la literatura, tardando menos de un segundo en los casos de la Clase 1 (donde los ítems son homogéneos) y menos de un minuto en los casos de la Clase 5 (donde los ítems son altamente heterogéneos). La diferencia entre las distancias alcanzadas es bastante pequeña, esto significa que la estructura de costos de la función objetivo incorporada en este trabajo, fuerza al algoritmo a balancear el peso de la carga con que atraviesa los arcos, cómo se ilustra en la Figura 1, gran parte de las aristas de una solución se conserva en la otra, modificando únicamente su sentido.

Por otro lado, en la Tabla 6.3 se presentan los resultados obtenidos al modificar la metodología para que busque minimizar la distancia total recorrida de las rutas. Estos resultados se comparan de forma justa con las mejores soluciones reportadas por los trabajos previos en la literatura [Leung et al. \(2013\)](#), [Dominguez et al. \(2016\)](#) y [Domínguez Rivero et al. \(2016\)](#). En especial, la columna BKS es la mejor solución conocida para cada instancia y la columna GAP es la diferencia porcentual entre las distancias del BKS y la obtenida en este trabajo. Cabe resaltar que se utilizaron los mismos parámetros establecidos para el 2L-FHFVRP. Los tiempos de cómputo alcanzados son similares a los obtenidos en la versión con combustible, no se reportan debido a que la comparación uno a uno no sería justa, debido a las diferencias entre arquitecturas de cómputo y desarrollo.

Id	Clase 1			Clase 2			Clase 3			Clase 4			Clase 5		
	BKS	Solución	GAP (%)	BKS	Solución	GAP (%)	BKS	Solución	GAP (%)	BKS	Solución	GAP (%)	BKS	Solución	GAP (%)
1	596,07	596,07	0	602,88	610,37	1,2	597,04	597,04	0	596,07	596,07	0	596,07	597,04	0,2
2	670,3	679,18	1,3	700,94	707,08	0,9	728,24	733,47	0,7	684,99	717,48	4,5	670,3	679,18	1,3
3	745,51	757,18	1,5	765,8	801,62	4,5	777,6	788,53	1,4	754,87	765,8	1,4	754,87	770,17	2
4	694,33	694,33	0	697,06	715,75	2,6	697,06	698,3	0,2	703,76	703,76	0	694,87	710,19	2,2
5	758,47	758,47	0	762,92	804,14	5,1	772,53	779,91	0,9	768,74	798,12	3,7	758,47	758,47	0
6	809,56	809,56	0	824,6	830,56	0,7	827,6	857,22	3,5	836,22	882,99	5,3	822,69	829,3	0,8
7	3211,53	3213,88	0,1	5952,09	6089,29	2,3	5135,66	5415,49	5,2	5466,9	5710,76	4,3	4723,64	4939,01	4,4
8	3184,45	3201,1	0,5	4804,14	4814	0,2	6400,11	6717,82	4,7	6290,15	6304,94	0,2	3864,97	3830,33	-0,9
9	1029,95	1031,55	0,2	1029,95	1089,67	5,5	1029,95	1037,79	0,8	1052,62	1084,6	2,9	1029,95	1029,95	0
10	4932,34	5082,18	2,9	7439,25	7888,78	5,7	6173,72	6360,87	2,9	7977,48	7879,66	-1,2	6065	6179,08	1,8
11	4932,34	5020,19	1,7	8007,51	8545,93	6,3	8289,87	8667,75	4,4	8820,13	8886,6	0,7	6053,52	6177,61	2
12	1655,73	1655,73	0	1674,56	1683,15	0,5	1681,51	1702,72	1,2	1680,01	1686,77	0,4	1677,52	1702,99	1,5
13	14579,87	14902,5	2,2	26318,79	27766	5,2	24598,85	25713,3	4,3	26103,64	27078,9	3,6	22808,22	23527,2	3,1
14	9097,52	9227,83	1,4	10389,98	10833,3	4,1	11345,99	11438,7	0,8	10233,7	10601,4	3,5	10005,07	10364,1	3,5
15	9097,52	9245,66	1,6	10780,29	10958,9	1,6	10753,2	11108,8	3,2	11449,4	12263,2	6,6	11527,11	12172,3	5,3
16	1276,22	1297,13	1,6	1280,9	1290,63	0,8	1288,07	1309,27	1,6	1290	1330,26	3	1280,9	1296,03	1,2
17	1764,32	1783,97	1,1	1756,5	1860,38	5,6	1766,5	1816,89	2,8	1766,5	1836,54	3,8	1768,57	1753,96	-0,8
18	3140,55	3203,28	2	5862,61	5882,47	0,3	5197,55	5479,62	5,1	6084,39	7334,08	17	4435,96	5416,18	18,1
19	1490,28	1566,72	4,9	3955,68	4797,48	17,5	3907,65	5209,65	25	4641,27	6027,38	23	3284,6	3775,79	13
20	1776,44	1870,38	5	5354,01	6547,4	18,2	5468,92	6194,45	11,7	5689,54	6894,8	17,5	4133,83	5103,29	19
21	2470,47	2576,9	4,1	6858,61	9597,62	28,5	8228,34	10261,5	19,8	7840,41	9592,66	18,3	5437,85	7768,4	30
22	2470,47	2566,33	3,7	7166,96	9463,43	24,3	8117,96	10259,2	20,9	8107,14	10217,6	20,7	5899,7	7229,97	18,4
23	2470,47	2600,85	5	7984,9	10190	21,6	7555,01	9747,97	22,5	8230,75	10190	19,2	5504,2	7503,28	26,6
24	2470,47	2580,33	4,3	4584,58	5561,92	17,6	4271,53	4800,68	11	4520,48	5050,71	10,5	3805,04	4243,31	10,3
25	2728,4	2961,77	7,9	10679,78	14814,3	27,9	10049,97	12714,1	21	11558,3	13917,1	16,9	7227,21	9408,58	23,2
26	3477,26	3587,97	3,1	10595,48	13400,8	20,9	10700,47	12833	16,6	11725,4	14424,9	18,7	8185,74	9234,58	11,4
27	2728,4	2958,89	7,8	5373,07	6536,98	17,8	5735,3	6326,48	9,3	5497,14	6202,87	11,4	5101,17	5640,89	9,6
28	4065,7	4281,62	5	19433,58	24899,1	22	20023,42	23315,4	14,1	21579,76	25619,3	15,8	16097,78	18793,9	14,3
29	9105,07	9897,04	8	20379,1	24508,5	16,8	20343,62	22334,8	8,9	21568,48	24508,5	12	20143,84	21399,6	5,9
30	3946,79	4289,22	8	14900,24	18742	20,5	15222,37	18742	18,8	16450,52	20148,9	18,4	10388,77	13369	22,3
31	5370,31	5836,01	8	18282,14	23119,8	20,9	20243,98	23516,9	13,9	21480,92	24618,5	12,7	14853,19	18341,8	19
32	5370,34	5860,83	8,4	18935,52	24364,8	22,3	17921,4	21408,4	16,3	20308,82	24038,8	15,5	13644,69	17064,7	20
33	5377,08		8,4	18846,63	24566,5	23,3	20344,29	23102,4	11,9	21899,41	24670,6	11,2	13648,79	16183,3	15,7
34	4263,78	4493,55	5,1	13606,24	15922,4	14,5	14185,86	15938,9	11	15326,26	16858,9	9,1	10480,51	11227,4	6,7
35	3911,17	4139,41	5,5	8340,24	9859,26	15,4	8876,94	10169,6	12,7	9226,53	10635,9	13,3	7776,65	8608,66	9,7
36	2847,02	3010,48	5,4	4290,8	5093,51	15,8	4524,41	5093,51	11,2	4280,03	4931,87	13,2	3883,48	4279,52	9,3
Promedio			3,49			11,64			8,9			9,37			9,16

Tabla 6.3: Resultados obtenidos por el GA-GRASP y comparación con las mejores soluciones publicadas en la literatura (BKS)

En la Tabla 6.3 se muestra que la metodología propuesta logra resolver todas las instancias de la literatura, alcanzado varias de las mejores soluciones reportadas y mejorando algunas. La diferencia entre las distancias alcanzadas es de 8,5 % en promedio. Cabe resaltar que, los algoritmos contra los que se compara buscan mejorar su función objetivo (minimizar las distancias totales) a través de operadores de búsqueda especializados. El algoritmo propuesto fue fácilmente adaptado para esta versión del problema de carga y empaquetamiento, conservando su estructura original.

6.7. Conclusiones y trabajos futuros

Se resolvió el problema integrado de enrutamiento de vehículos y carga de mercancías considerando reducir el consumo de combustible, a través de un algoritmo genético elitista, de fácil implementación, una característica básica de las metodologías aproximadas. Se obtuvieron resultados satisfactorios en tiempos de cómputo razonables (menos de 30 segundos en promedio). Su comportamiento fue validado comparando su desempeño con los mejores trabajos previos publicados, para una variante del problema que no considera el consumo de combustible, se alcanzaron varias de las mejores soluciones conocidas, se mejoraron algunas instancias y se mantuvo una brecha no muy lejana a los mejores resultados reportados.

Se presentó una adaptación eficiente y rápida de la codificación *split*, que permite incorporar fácilmente las restricciones de empaquetamiento, evitando aumentar los tiempos de cómputo vertiginosamente. El uso de esta representación permite introducir restricciones reales del problema sin modificar la estructura principal del algoritmo, dado que las consideraciones prácticas se validan en la creación del grafo auxiliar, restricciones como ventanas de tiempo, duración de las rutas, entre otros, se limitan a crear o no los arcos de las rutas válidas.

Se adaptó el algoritmo GRASP reactivo para la validación de rutas factibles por empaquetamiento, se utilizó la codificación de espacios residuales máximos para considerar la restricción de carga secuencial, de forma tal que, entre clientes se conservan únicamente los espacios factibles a la hora del cargue. Además de esto, su estructura ligera hace que no

dependa de un gran número de parámetros y se ajuste a la complejidad del surtido de los ítems.

Como trabajo futuro se propone comparar el estudio aquí presentado, contra metodologías que alcanzan buenos resultados, para otras variantes del problema de rutas de vehículos y empaquetamiento de mercancías, en especial, se sugiere implementar el algoritmo de trayectoria tipo ILS, el cual en la literatura aparece con mejor desempeño. Además de esto, se recomienda utilizar y adaptar la metodología propuesta a problemas que consideren restricciones reales de la industria, incorporando condiciones como: ventanas de tiempo, duración máxima de viaje, vehículos eléctricos, mercancía apilable, entre otros.

Capítulo 7

Conclusiones, trabajos futuros y producción

7.1. Conclusiones

Al final del desarrollo de este trabajo, se aporta al estado del arte el estudio de dos problemas nuevos, un modelo matemático y metodologías computacionales adecuadas para abordar variantes de enrutamiento que analizan de manera realista la carga asociada a las demandas de los clientes.

Se contribuyen dos metodologías nuevas para el problema 3L-CVRP; ambas utilizan hibridación tanto metaheurística como matheurística, explorando abordajes desde diferentes perspectivas para sumar en el estudio de este problema compuesto por dos problemas NP-duros articulados.

Adicionalmente, para fortalecer un enfoque más realista de los problemas de enrutamiento de vehículos, se hace un estudio específico para el enrutamiento de vehículos con consideraciones verdes o ambientales. Al realizar este estudio, se propone un nuevo problema, el cual, en efecto, fue formalizado a través de un nuevo modelo matemático que se aporta al estado del arte correspondiente.

La inmersión en problemas de enrutamiento de vehículos con consideraciones ambientales, muestra que medir el impacto ecológico, no sólo aporta en la disminución de las consecuencias de las actividades de transporte realizadas por la industria, si no que también genera un desarrollo importante en la forma de modelar y resolver los problemas de enrutamiento de vehículos por todos los elementos realistas que hay que agregar a las representaciones computacionales; sumando en la articulación de la investigación con las aplicaciones en el sector productivo.

La claridad adquirida en el estudio presentado de enrutamiento verde, al involucrar elementos como especificaciones técnicas del vehículo como su peso, resistencia al viento, consumo energético relacionado con su velocidad de desplazamiento y tiempo de ruta costado con la mano de obra del conductor encargado de la distribución de los bienes; sentó las bases para encontrar la forma adecuada de involucrar impacto ambiental en una variante de enrutamiento empaquetamiento realista.

Se presenta entonces un problema no estudiado, insinuado como posible variante en la literatura especializada previa, que integra todos los estudios realizados en el desarrollo de la investigación. El problema del 2L-FHFVRP, es decir, demandas no apilables suplidas con una flota de vehículos heterogénea y considerando impacto ambiental minimizando el consumo de combustible de las rutas que realizan la distribución de los bienes, fue resuelto además con la extensión de una codificación clásica para problemas de enrutamiento de vehículos, brindando una forma de representar este problema realista para futuras investigaciones.

Durante esta investigación, fueron seguidas dos tendencias importantes identificadas en el estado del arte: (i) aunque sumar variantes al problema de enrutamiento de vehículos es muy valioso para este campo de estudio, hay un interés creciente en resolver eficientemente los problemas de agente viajero y cartero chino específicos embebidos en los problemas de enrutamiento complejos, dado que resolverlos de la mejor manera posible en términos computacionales, mejora directamente la solución de problemas de enrutamiento correspondientes, y de porte cercano a los que se presentan en la práctica. Y (ii) en la misma línea de resolver el TSP que sigue las reglas del problema de enrutamiento correspondiente,

junto con el notable avance de los recursos computacionales como el hardware y librerías especializadas de alto rendimiento, tornan de gran interés los algoritmos clásicos que afrontan la manipulación y toma de decisiones sobre los grafos que surgen en el proceso de solución; en este trabajo se utilizaron algoritmos propuestos hace varias décadas para el core de las implementaciones (especialmente la variante con impacto ambiental), y su eficiencia computacional es notable, dado que fueron propuestos en tiempos donde las plataformas de cómputo eran muy limitadas.

7.2. Trabajos Futuros

A continuación se enumeran algunos de los posibles trabajos futuros que se podrían desprender de esta tesis doctoral:

1. Dado que se realizó el estudio para 2L-HFVRP y 3L-CVRP, se podría abordar la variante 3L-HFVRP, la cual se encuentra poco estudiada. La adaptación de los algoritmos desarrollados en esta investigación a este nuevo contexto, sería un aporte importante en esta familia de problemas.
2. Proponer un modelo matemático para el 3L-HFVRP.
3. Proponer nuevas metodologías de solución para el 3L-HFVRP.
4. En este trabajo se propuso el 2L-FHFVRP, una variante nueva (detallada en el documento). Se observó que las únicas instancias disponibles, diseñadas para el mismo problema sin incluir el consumo de combustible, pueden ser mejoradas para que exista una diferenciación mayor entre los pesos de las demandas de los clientes, y de esta manera, tener algunos casos dentro de la librería que presenten una correlación mayor entre el consumo de combustible y los bienes transportados.
5. Proponer el problema 3L-FHFVRP que aún no ha sido estudiado y aportar: instancias apropiadas, modelo o modelos matemáticos y metodologías de solución.

6. Proponer y estudiar nuevas variantes de enrutamiento-empaquetamiento que incluyan otras variables que no hayan sido tenidas en cuenta en los estudios previos y que estén presentes en las aplicaciones reales.

El desarrollo de los trabajos enumerados pueden ser realizados en investigaciones de maestría y doctorado, generando aportes a la comunidad académica.

7.3. Producción

Las publicaciones alcanzadas durante esta investigación se listan a continuación:

- **Memorias de Congreso IEEE:** Escobar, L. M., Martínez, D. Á., Escobar, J. W., Linfati, R., Mauricio, G. E. (2015, October). A hybrid metaheuristic approach for the capacitated vehicle routing problem with container loading constraints. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 1374-1382). IEEE.
- **Publicación en Revista Especializada A1(Pubindex):** Escobar-Falcón, L. M., Álvarez-Martínez, D., Granada-Echeverri, M., Escobar, J. W., Romero-Lázaro, R. A. (2016). A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3L-CVRP). *Revista Facultad de Ingeniería Universidad de Antioquia*, (78), 09-20.
- **Capítulo de Libro:** Cacchiani, V., Contreras-Bolton, C., Escobar, J. W., Escobar-Falcon, L. M., Linfati, R., Toth, P. (2018). An Iterated Local Search Algorithm for the Pollution Traveling Salesman Problem. In *New Trends in Emerging Complex Real Life Problems* (pp. 83-91). Springer, Cham.
- **Publicación en Revista Especializada Q1:** Moreno, C., Falcón, L., Bolaños, R., Subramanian, A., Zuluaga, A., Echeverri, M. (2019). A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. *International Journal of Industrial Engineering Computations*, 10(3), 361-374.

- **Publicación en Revista Especializada Q3:** Moreno, C., Falcón, L., Escobar, J., Zuluaga, A., Echeverri, E. (2019). Heuristic constructive algorithm for work-shift scheduling in bus rapid transit systems. *Decision Science Letters*, 8(4), 519-530.

Las publicaciones presentadas corresponden tanto a los trabajos realizados dentro de esta investigación, junto con productos que se generaron al realizar experimentos con las técnicas aquí desarrolladas en el contexto de scheduling o asignación de tareas.

Adicionalmente, se tienen dos trabajos en revisión al momento del cierre de esta tesis:

- **Revista Especializada Optimization Letters Q2:** Matheuristic Algorithms for the Pollution Traveling Salesman Problem.
- **Corrección de Estilo Idioma Inglés:** Algoritmo genético especializado para el problema de enrutamiento de vehículos con flota heterogénea y restricciones de empaquetamiento bidimensional minimizando el consumo de combustible

Bibliografía

- Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. 1988.
- Murat Albayrak and Novruz Allahverdi. Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms. *Expert Systems with Applications*, 38(3):1313–1320, 2011. ISSN 09574174.
- Maria Teresa Alonso, Ramon Alvarez-Valdes, Jose Manuel Tamarit, and Francisco Parreño. A reactive grasp algorithm for the container loading problem with load-bearing constraints. *European Journal of Industrial Engineering*, 8(5):669–694, 2014.
- Sia Ardekani, Ezra Hauer, and Bahram Jamei. Traffic impact models. *Chapter 7 in Traffic Flow Theory, Oak Bridge National Laboratory Report*, 1992.
- A. Attanasio, A. Fuduli, G. Ghiani, and C. Triki. Integrated shipment dispatching and packing problems: a case study. *Journal of Mathematical Modelling and Algorithms*, 6: 77–85, 2007.
- R. Baldacci, P. Toth, and D. Vigo. *Exact algorithms for routing problems under vehicle capacity constraints*, volume 175. Annals of Operations Research, Springer US, 2010.
- Ronald H. Ballou. *Bussines Logistics/Supply Chain Management*. 2004.
- W Banzhaf. The “molecular” traveling salesman. *Biological Cybernetics*, 64(1):7–14, 1990. ISSN 0340-1200.
- Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

- A. Bortfeldt. A hybrid algorithm for the capacitated vehicle routing problem with threedimensional loading constraints. *Computers & Operations Research*, 39:2248–2257, 2012.
- A. Bortfeldt and J. Homberger. Packing first, routing second - a heuristic for the vehicle routing and loading problem. *Computers & Operations Research*, 40:873–885, 2013.
- PG Boulter, Ian S McCrae, and Tim J Barlow. *A review of instantaneous emission models for road vehicles*. TRL Limited, 2007.
- J. Brandao. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operation Research*, 195:716–728, 2009.
- Luciana Buriol, Paulo M. França, and Pablo Moscato. A new memetic algorithm for the asymmetric traveling salesman problem. *Journal of Heuristics*, 10(5):483–506, September 2004. ISSN 1381-1231.
- Valentina Cacchiani, Carlos Contreras-Bolton, John W Escobar, Luis M Escobar-Falcon, Rodrigo Linfati, and Paolo Toth. An iterated local search algorithm for the pollution traveling salesman problem. In *New Trends in Emerging Complex Real Life Problems*, pages 83–91. Springer, 2018.
- Guillermo Alberto Camacho, David Alvarez, and Daniel Cuellar. Heuristic approach for the multiple bin-size bin packing problem. *IEEE Latin America Transactions*, 16(2):620–626, 2018.
- Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.
- Paul C Chu and John E Beasley. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24(1):17–23, 1997.
- Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

- Ryan Conrad and Miguel Figliozzi. Algorithms to quantify impact of congestion on time-dependent real-world urban freight distribution networks. *Transportation Research Record: Journal of the Transportation Research Board*, (2168):104–113, 2010.
- Carlos Contreras-Bolton and Victor Parada. Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. *PloS one*, 10(9):e0137724, 2015.
- Jean-François Côté, Gianfranco Guastaroba, and Maria Grazia Speranza. The value of integrating loading and routing. *European Journal of Operational Research*, 257(1):89–105, 2017.
- S. Cullinane and J. Edwards. Assessing the environmental impacts of freight transport. In A. McKinnon, S. Cullinane, Browne M, and A. Whiteing, editors, *Green Logistics*, chapter 2, pages 31–48. Kogan Page, London, 2010.
- C. D’Ambrosio, A. Lodi, and S. Martello. *Combinatorial traveling salesman problem algorithms*. Wiley Encyclopedia of Operations Research and Management Science, 2011.
- George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.
- George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012.
- Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.
- Oscar Dominguez, Angel A Juan, Barry Barrios, Javier Faulin, and Alba Agustin. Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, 236(2):383–404, 2016.

- Oscar L Domínguez Rivero, Angel A Juan Pérez, Ignacio A de la Nuez Pestana, and Djamila Ouelhadj. An ils-biased randomization algorithm for the two-dimensional loading hfvrrp with sequential loading and items rotation. *Journal of the Operational Research Society*, 67(1):37–53, 2016.
- C. Duhamel, P. Lacomme, A. Quilliot, and H. Toussaint. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38:617–640, 2011.
- Jr E.G. Coffman, J. Csirik, G. Galambos, S. Martello, and D. Vigo. *Handbook of Combinatorial Optimization*, chapter Bin Packing Approximation Algorithms: Survey and Classification. Springer, segunda edition, 2013.
- Richard Eglese and Daniel Black. Optimizing the routeing of vehicles. In A. McKinnon, S. Cullinane, Browne M, and A. Whiteing, editors, *Green Logistics*, chapter 10, pages 223–235. Kogan Page, London, 2 edition, 2012.
- Richard Eglese, Will Maden, and Alan Slater. A road timetableTM to aid vehicle routing and scheduling. *Computers & operations research*, 33(12):3508–3519, 2006.
- A. E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Springer Publishing Company, Incorporated, 2nd edition, 2015. ISBN 3662448734, 9783662448731.
- Ágoston Endre Eiben, R. Hinterding, and Zbigniew Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, July 1999. ISSN 1089778X.
- Ágoston Endre Eiben, Zbigniew Michalewicz, M Schoenauer, and Jim E Smith. Parameter Control in Evolutionary Algorithms. In FernandoG. Lobo, CláudioF. Lima, and Zbigniew Michalewicz, editors, *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*, pages 19–46. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-69431-1.
- John Willmer Escobar, Rodrigo Linfati, and Paolo Toth. A two-phase hybrid heuristic

- algorithm for the capacitated location-routing problem. *Computers & Operations Research*, 40(1):70–79, 2013.
- John Willmer Escobar, Rodrigo Linfati, Maria G Baldoquin, and Paolo Toth. A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological*, 67:344–356, 2014a.
- John Willmer Escobar, Rodrigo Linfati, Paolo Toth, and Maria G Baldoquin. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of Heuristics*, 20(5):483–509, 2014b.
- Luis Miguel Escobar, David Álvarez Martínez, John Willmer Escobar, Rodrigo Linfati, and Granada E Mauricio. A hybrid metaheuristic approach for the capacitated vehicle routing problem with container loading constraints. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, pages 1374–1382. IEEE, 2015.
- Luis Miguel Escobar-Falcón, David Álvarez-Martínez, Mauricio Granada-Echeverri, John Willmer Escobar, and Rubén Augusto Romero-Lázaro. A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3l-cvrp). *Revista Facultad de Ingeniería Universidad de Antioquia*, (78):09–20, 2016.
- A Esteves-Booth, T Muneer, J Kubie, and H Kirby. A review of vehicular emission models and driving cycles. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 216(8):777–797, 2002.
- EU-27-European-Union. Member countries of the eu (year of entry), 2013. URL https://europa.eu/european-union/about-eu/countries_en.
- T. Feo and M. G. C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.
- Miguel Figliozzi. Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board*, (2197):1–7, 2010.

- Miguel Andres Figliozi. The impacts of congestion on time-definitive urban freight distribution networks co 2 emission levels: Results from a case study in portland, oregon. *Transportation Research Part C: Emerging Technologies*, 19(5):766–778, 2011.
- Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.
- David J. Forkenbrock. External costs of intercity truck freight transportation. *Transportation Research Part A: Policy and Practice*, 33(7–8):505 – 526, 1999. ISSN 0965-8564. doi: [http://dx.doi.org/10.1016/S0965-8564\(98\)00068-8](http://dx.doi.org/10.1016/S0965-8564(98)00068-8). URL [//www.sciencedirect.com/science/article/pii/S0965856498000688](http://www.sciencedirect.com/science/article/pii/S0965856498000688).
- David J. Forkenbrock. Comparison of external costs of rail and truck freight transportation. *Transportation Research Part A: Policy and Practice*, 35(4):321 – 337, 2001. ISSN 0965-8564. doi: [http://dx.doi.org/10.1016/S0965-8564\(99\)00061-0](http://dx.doi.org/10.1016/S0965-8564(99)00061-0). URL [//www.sciencedirect.com/science/article/pii/S0965856499000610](http://www.sciencedirect.com/science/article/pii/S0965856499000610).
- Anna Franceschetti, Dorothée Honhon, Tom Van Woensel, Tolga Bektaş, and Gilbert Laporte. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.
- G. Fuellerer, K. F. Doerner, R. Hartl, and M. Iori. Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 36: 655–673, 2009.
- G. Fuellerer, K.F. Doerner, R. Hartl, and M. Iori. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201: 751–759, 2010.
- M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51:4–18, 2007.
- M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40:342–350, 2006.

- B. Golden, S. Raghavan, and E. Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges.*, volume 43. Operations Research/Computer Science Interfaces Series, 2008.
- Michael D Grant, Anne Choate, and Lauren Pederson. Assessment of greenhouse gas analysis techniques for transportation projects. Technical report, 2008.
- John J. Grefenstette, Rajeev Gopal, Brian J. Rosmaita, and Dirk Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 160–168, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc. ISBN 0-8058-0426-9.
- Carlos Groba, Antonio Sartal, and Xosé H Vázquez. Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices. *Computers & Operations Research*, 56:22–32, 2015.
- Daniel Guimarans, Oscar Dominguez, Angel A Juan, and Enoc Martinez. A multi-start simheuristic for the stochastic two-dimensional vehicle routing problem. In *2016 Winter simulation conference (Wsc)*, pages 2326–2334. IEEE, 2016.
- Daniel Guimarans, Oscar Dominguez, Javier Panadero, and Angel A Juan. A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory*, 89:1–14, 2018.
- K. Hamdi-Dhaoui, N. Labadie, and A. Yalaoui. *IJCCI 2012 - Proceedings of the 4th International Joint Conference on Computational Intelligence*, chapter Memetic algorithm with population management for the two-dimensional loading vehicle routing problem with partial conflicts, pages 189–195. SciTePress, 2012.
- John Hickman, Dieter Hassel, Robert Joumard, Zissis Samaras, and S Sorenson. Methodology for calculating transport emissions and energy consumption. 1999.
- M. Iori. Metaheuristic algorithms for combinatorial optimization problems. *4OR*, 3:163–166, 2005.

- M. Iori, J.J. Salazar González, and D. Vigo. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41:253–264, 2007.
- Manuel Iori. An annotated bibliography of combined routing and loading problems. *Yugoslav Journal of Operations Research*, 23(3), 2013.
- O Jabali, T Woensel, and AG de Kok. Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012.
- L. Junqueira, J.F. Oliveira, M.A. Carravilla, and R. Morabito. An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research*, 20:645–666, 2013.
- Imdat Kara, Bahar Y Kara, and M Kadri Yetis. Energy minimizing vehicle routing problem. In *International Conference on Combinatorial Optimization and Applications*, pages 62–71. Springer, 2007.
- S. Khebbache-Hadji, C. Prins, A. Yalaoui, and M. Reghioui. Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. *Central European Journal of Operations Research*, 21:307–336, 2013.
- R. Knight. Mobility 2030: Meeting the challenges to sustainability. Technical report, University of Zurich, Department of Informatics, Hertfordshire, London, 2004. URL <http://wbcsdpublications.org/project/mobility-2030-meeting-the-challenges-to-sustainability-executive-summary-2004/>.
- G. Koloch and B. Kaminski. Nested vs. joint optimization of vehicle routing problems with three-dimensional loading constraints. *Engineering Letters*, 18:193–198, 2010.
- Raphael Kramer, Anand Subramanian, Thibaut Vidal, and F Cabral Lucídio dos Anjos. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015.

- Yiyo Kuo. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1):157–165, 2010.
- Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009. doi: 10.1287/trsc.1090.0301. URL <http://dx.doi.org/10.1287/trsc.1090.0301>.
- Pedro Larranaga, Cindy M. H. Kuijpers, Roberto H. Murga, Inaki Inza, and Sejla Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.
- A.N. Letchford and A. Lodi. *Mathematical programming approaches to the traveling salesman problem*. Wiley Encyclopedia of Operations Research and Management Science, 2011.
- S.C.H. Leung, J. Zheng, D. Zhang, and X. Zhou. Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services and Manufacturing Journal*, 22:61–82, 2010.
- S.C.H. Leung, X. Zhou, D. Zhang, and J. Zheng. Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 38:205–215, 2011.
- S.C.H. Leung, Z. Zhang, D. Zhang, X. Hua, and M.K. Lim. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 225:199–210, 2013.
- Ke Li, Alvaro Fialho, Sam Kwong, and Qingfu Zhang. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 18(1):114–130, 2014.
- Canhong Lin, King Lun Choy, George TS Ho, Sai Ho Chung, and HY Lam. Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, 41(4):1118–1138, 2014.
- Shuguang Liu, Weilai Huang, and Huiming Ma. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):434–445, 2009.

- A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141:3–13, 2002.
- A. Lodi, S. Martello, M. Monaci, and D. Vigo. *Paradigms of Combinatorial Optimization: Problems and New Approaches*, chapter Two-dimensional bin packing problems, pages 107–129. ISTE and John Wiley & Sons, 2010.
- Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search: Framework and applications. In *Handbook of metaheuristics*, pages 363–397. Springer, 2010.
- Will Maden, Richard Eglese, and Dan Black. Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61(3): 515–522, 2010.
- D Alvarez Martínez, R Alvarez-Valdes, and F Parreño. A grasp algorithm for the container loading problem with multi-drop constraints. *Pesquisa Operacional*, 35(1):1–24, 2015.
- L. Martínez and C.A. Amaya. A vehicle routing problem with multi-trips and time windows for circular items. *Journal of the Operational Research Society*, 2013.
- A. McKinnon. Environmental sustainability: A new priority for logistics managers. In A. McKinnon, S. Cullinane, Browne M, and A. Whiteing, editors, *Green Logistics*, chapter 1, pages 3–30. Kogan Page, London, 2010.
- L. Miao, Q. Ruan, K. Woghiren, and Q. Ruo. A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. *RAIRO - Operations Research*, 46:63–82, 2012.
- Guido JL Micheli and Fabio Mantella. Modelling an environmentally-extended inventory routing problem with demand uncertainty and a heterogeneous fleet under carbon control policies. *International Journal of Production Economics*, 204:316–327, 2018.
- Chiung Moon, Jongsoo Kim, Gyunghyun Choi, and Yoonho Seo. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 140(3):606–617, 2002.

- A. Moura. *Intelligent Decision Support*, chapter A multi-objective genetic algorithm for the vehicle routing with time windows and loading, pages 187–201. 2008.
- A. Moura and J. F. Oliveira. An integrated approach to vehicle routing and container loading problems. *OR Spectrum*, 31:775–800, 2009.
- Leonidas Ntziachristos, Zissis Samaras, S Eggleston, N Gorissen, D Hassel, AJ Hickman, et al. Copert iii. *Computer Programme to calculate emissions from road transport, methodology and emission factors (version 2.1)*, European Energy Agency (EEA), Copenhagen, 2000.
- Manfred Padberg and Giovanni Rinaldi. An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming*, 47(1):19–36, 1990.
- Andrew Palmer. The development of an integrated routing and carbon dioxide emissions model for goods vehicles. 2007.
- F. Parreno, R. Alvarez-Valdés, J. M. Tamarit, and J. F. Oliveira. Neighborhood structures for the container loading problem: A vns implementation. *Journal of Heuristics*, 161:1–22, 2010.
- Vangelis Th Paschos. *Paradigms of combinatorial optimization: problems and new approaches*, volume 2. John Wiley & Sons, 2013.
- Hanne Pollaris, Kris Braekers, An Caris, Gerrit K Janssens, and Sabine Limbourg. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330, 2015.
- Jean-Yves Potvin. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 63(3):337–370, 1996.
- C. Prins. Two memetic algorithms for heterogeneous fleet vrp. *Engineering Applications of Artificial Intelligence*, 22:916–928, 2009.
- Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers Operations Research*, 31(12):1985 – 2002, 2004. ISSN 0305-0548.

doi: [http://dx.doi.org/10.1016/S0305-0548\(03\)00158-8](http://dx.doi.org/10.1016/S0305-0548(03)00158-8). URL [//www.sciencedirect.com/science/article/pii/S0305054803001588](http://www.sciencedirect.com/science/article/pii/S0305054803001588).

Jiani Qian and Richard Eglese. Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, 248(3):840–848, 2016.

Bernd Reisleben, Peter Merz, and B Freisleben. A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 616–621, May 1996. ISBN 0780329023.

M.G.C. Resende and C.C. Ribeiro. *Handbook of Metaheuristics*, chapter Greedy Randomized Adaptive Search Procedures, pages 219–249. Kluwer, 2003.

Oscar L Domínguez Rivero, Angel A Juan Pérez, Ignacio A de la Nuez Pestana, and Djamila Ouelhadj. An ils-biased randomization algorithm for the two-dimensional loading hfvpr with sequential loading and items rotation. *Journal of the Operational Research Society*, 67(1):37–53, 2016.

Q. Ruan, Z. Zhang, L. Miao, and H. Shen. A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 40:1579–1589, 2013.

Nikola Ruzinski, Natalija Koprivanac, Slaven Dobrovic, Gordana Stefanovic, Gilberto Tavares, Zdena Zsigraiova, Viriato Semiao, and Maria da Graça Carvalho. A case study of fuel savings through optimisation of MSW transportation routes. *Management of Environmental Quality: An International Journal*, 19(4):444–454, 2008.

V. Schmid, K.F. Doerner, and G. Laporte. Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224:435–448, 2013.

Y. Shen and T. Murata. Pick-up scheduling of two-dimensional loading in vehicle routing problem by using ga. In *In Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS 2012*, volume 2, pages 1532–1537, 2012.

- Lawrence V Snyder and Mark S Daskin. A random-key genetic algorithm for the generalized traveling salesman problem. *European journal of operational research*, 174(1):38–53, 2006.
- J. Strodl, K.F. Doerner, F. Tricoire, and R.F. Hartl. *Hybrid Metaheuristics, Lecture Notes in Computer Science*, volume 6373, chapter On index structures in hybrid metaheuristics for routing problems with hard feasibility checks: An application to the 2-dimensional loading vehicle routing problem, pages 160–173. Springer Berlin Heidelberg, 2010.
- Anand Subramanian, P Penna, Eduardo Uchoa, and Luiz Satoru Ochi. A hybrid algorithm for the fleet size and mix vehicle routing problem. In *International Conference on Industrial Engineering and Systems Management*, pages 414–427, 2011.
- Anand Subramanian, Eduardo Uchoa, and Luiz Satoru Ochi. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531, 2013.
- Yoshinori Suzuki. A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, 16(1):73–77, 2011.
- Gilbert Syswerda. Schedule optimization using genetic algorithms. In Lawrence Davis, editor, *Handbook of Genetic Algorithms*, pages 332–349, New York, 1991. Van Nostrand Reinhold.
- Y. Tao and F. Wang. A new packing heuristic based algorithm for vehicle routing problem with three-dimensional loading constraints. In *Conference on Automation Science and Engineering In 2010 IEEE International*, 2010.
- Yi Tao and Fan Wang. An effective tabu search approach with improved loading algorithms for the 3l-cvrp. *Computers Operations Research*, 55:127 – 140, 2015. ISSN 0305-0548. doi: <http://dx.doi.org/10.1016/j.cor.2013.10.017>. URL <http://www.sciencedirect.com/science/article/pii/S0305054813003122>.
- C. D. Tarantilis, E. E. Zachariadis, and C. T. Kiranoudis. A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, 10:255–271, 2009.
- P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15:333–346, 2003.

- P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 2013.
- Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.
- Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.
- S Ubeda, FJ Arcelus, and J Faulin. Green logistics at eroski: A case study. *International Journal of Production Economics*, 131(1):44–51, 2011.
- US Environmental Protection Agency (US-EPA). Inventory of us greenhouse gas emissions and sinks: 1990–2008, 2010.
- A. S. Vicente. *Laying the foundations for greener transport — TERM 2011: transport indicators tracking progress towards environmental targets in Europe*. European Environment Agency, Copenhagen, Denmark, 2011, 2011. ISBN 978-92-9213-230-9. doi: 10.2800/82592.
- G. Wascher, H. Haubner, and H. Shchumann. A typology and classification of the 0most important packing problems in one and higher dimensions. *European Journal of Operational Research*, 183:1109–1130, 2007.
- Yiyong Xiao, QiuHong Zhao, Ikou Kaku, and Yuchun Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7):1419–1431, 2012.
- Shuai Yuan, Bradley Skinner, Shoudong Huang, and Dikai Liu. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. *European Journal of Operational Research*, 228(1):72–82, 2013.
- E. E. Zachariadis, C. D. Tarantillis, and C. T. Kiranoudis. A guided tabu search for the vehicle routing problema with two dimensional loading constraints. *European Journal of Operational Research*, 195:729–743, 2009.

- Zhenzhen Zhang, Lijun Wei, and Andrew Lim. An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological*, 82:20–35, 2015.
- W. Zhu, H. Qin, A. Lim, and L. Wang. A two-stage tabu search algorithm with enhanced packing heuristics for the 3l-cvrp and m3l-cvrp. *Computers & Operations Research*, 39: 2178–2195, 2012.